Manual de Usuario de Dr. Geo



Hilaire Fernandes, Andrea Centomo
Adrian Soto
OFSET
http://www.ofset.org

29 de agosto de 2004

Índice general

1. Introducción **7**
1.1. Preliminares 7
1.2. Dr. Geo en la red 9

2. Funciones Básicas **11**
2.1. Herramientas de Construcción 11
2.1.1. Herramientas de puntos 12
2.1.2. Herramientas de líneas 13
2.1.3. Herramientas de Transformación 14
2.1.4. Herramientas Numéricas 15
2.1.5. Herramientas para Macros de construcciones 17
2.2. Otras herramientas 17
2.2.1. Árbol lógico de construcción 17
2.2.2. Mover la figura 17
2.2.3. Borrar un objeto 17
2.2.4. Editar la Apariencia de un Objeto 18
2.2.5. Editando Propiedades de Objetos 20
2.2.6. Activar un enrejado 21
2.3. Preferencias de Usuario 21
2.3.1. Comportamiento Predeterminado 21
2.3.2. Otras Preferencias 22

3. Características Avanzadas **23**
3.1. Macros de Construcciones 23
3.1.1. Crear una Macro 24
3.1.2. Ejecutar una Macro 26
3.2. Script Scheme de Dr. Geo 29
3.2.1. DGS a través de ejemplos 29
3.2.2. Métodos de referencia para los scripts de Dr. Geo 33
3.3. Figura Scheme de Dr. Geo 36
3.3.1. Algunos Ejemplos 36
3.3.2. Métodos de referencia para las figuras Scheme de Dr. Geo 38
3.3.3. Sinónimos en DGS 46
3.3.4. Galería de Ejemplos 48
3.4. Bloquear las herramientas de la interfaz 51
3.4.1. Bloqueo de las herramientas 51
3.4.2. Desbloqueo de las herramientas 52

4. Archivos y documentos **53**
4.1. Guardar una construcción 53
4.2. Guardar una sesión 53
4.3. Guardar una Macro 54
4.4. Abrir un archivo 54

4.5. Exportar una figura 54
4.5.1. Exportar a LaTeX 54
4.5.2. Exportar a PostScript 55
4.5.3. Exportar a PNG 55
4.5.4. Exportar Fly Draw 55
4.5.5. Definir el área de exportación 55

5. Aplicaciones Didácticas **57**
5.1. Pitágoras y scripts 57
5.2. Teoremas y conjeturas 58
5.3. Números irracionales 60
5.4. Espiral de Baravelle 61
5.5. Cadena de Papus 65
5.6. Cálculo de π 66

6. Libro de recetas **69**
6.1. Crear un polígono regular 69
6.1.1. El modelo del polígono 69
6.1.2. La macro del polígono 70
6.1.3. Algunas consideraciones 70
6.2. Imprimir una figura 70
6.3. Colocar un párrafo de texto en la figura 71

A. Histórico. **77**

B. GNU Free Documentation License **79**

Copyright

Capítulo 1

Introducción

1.1. Preliminares

DR. GEO es un programa tanto de geometría interactiva como de programación en el lenguaje Scheme. Permite crear figuras geométricas, así como manipularlas interactivamente respetando sus restricciones geométricas. Ofrece igualmente la posibilidad de introducirse gradualmente en la programación. Es útil para la enseñanza a estudiantes de nivel básico o superior.

La interfaz de usuario de DR. GEO ha sido concebida para dar, dentro de un conjunto armónico, a la vez sencillez en su operación, ergonomía y funcionalidades avanzadas.

Así la interfaz de DR. GEO , con una gran sencillez, permite al neófito la familiarización con las funciones básicas del programa. Posteriormente, a medida que lo conozca más, el usuario conocerá los aspectos más avanzados de la interfaz y del funcionamiento de DR. GEO : múltiples modos para la construcción de objetos [1], macros de construcciones, sesiones, adaptabilidad de la interfaz, uso de scripts y de Figuras Scheme de DR. GEO(i.e. DR. GEO es programable en dos modos distintos). Estas funciones avanzadas generan muy poca sobrecarga a la interfaz, por lo que DR. GEO es utilizado con entusiasmo en la enseñanza primaria, lo que no resta su uso en el nivel medio.

En las secciones siguientes, expondremos las herramientas básicas. Posteriormente las funciones avanzadas serán presentadas en detalle.

La estructura de la interfaz es la siguiente :

1. la *barra de menú* característica con `Archivo`, `Editar`, `Macros`, `Ventanas`, `Ayuda` ;
2. la *barra de acciones* para crear una nueva figura o un texto explicativo; contiene también las herramientas de hacer/deshacer y la rejilla .

Para crear una figura geométrica nueva, el usuario debe hacer click en el primer botón de la barra de herramientas. De manera alternativa, una nueva figura puede ser creada usando el elemento de la barra del menú `Archivo->Nuevo->Figura`. Cuando una nueva figura es creada, una nueva *barra de íconos* formada por seis iconos aparece. Esta barra de íconos termina con un menú que permite ajustar la escala de la figura geométrica actual.

Además, una barra de íconos vertical, a la izquierda de la figura, ofrece un rápido acceso a las herramientas más utilizadas.

Los seis iconos de la barra de íconos son entradas de menú generales desde donde se llaman a funciones específicas. Estas funciones son descritas en el siguiente capítulo.

[1] Se trata de poder, a partir de un mismo comando, crear un tipo de objeto según se requiera. Por ejemplo, a partir del comando para la construcción de un círculo, el usuario puede crear un círculo a partir de su centro y de ya sea un punto, una longitud, un segmento, etc. Con todo, este comando es representado por un solo icono, y DR. GEO se anticipa para ofrecer al usuario la construcción deseada. El efecto inmediato es que existe una disminución en el tiempo de aprendizaje de la interfaz, al proponer un número importante de modos de construcción

Figura 1.1: Pantalla de Bienvenida de Dr. Geo

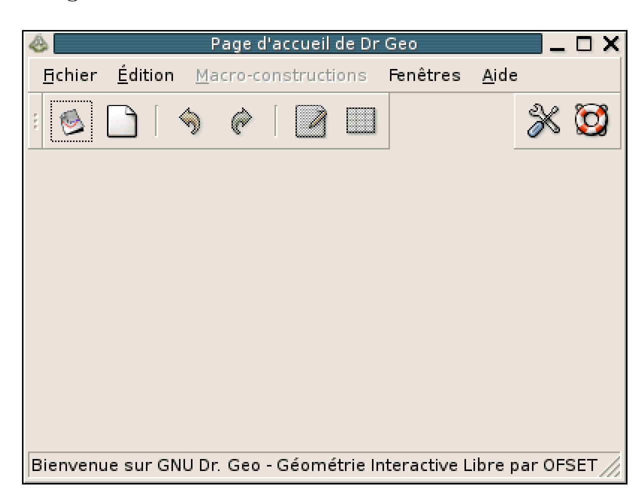
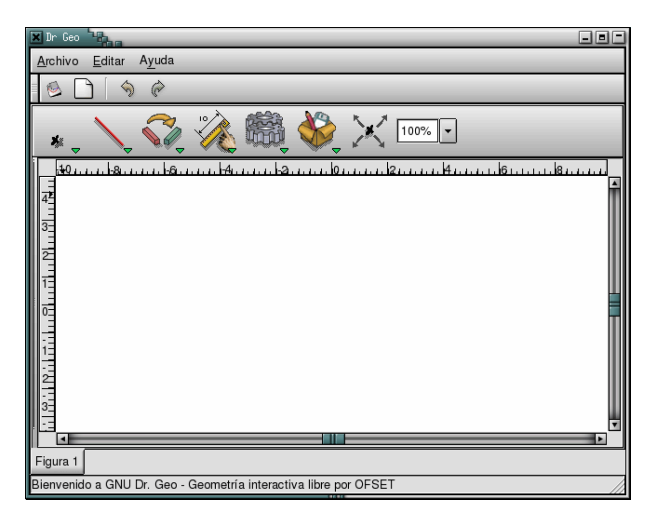


Figura 1.2: Una figura geométrica vacía



Para cada figura, un *árbol sinóptico* está disponible. Inicialmente este panel está oculto en el extremo izquierdo de modo que sólo la representación de la figura es visible. En cualquier momento, el usuario puede empujar el panel hacia la derecha como sigue: Lleve el ratón a la parte izquierda de la figura, justo a la izquierda de donde está la regla; cuando el puntero del ratón cambie de forma, presione el botón y continúe presionando mientras mueve el ratón a la derecha. Con ello, será visible la descripción de la figura. La descripción de una figura es un árbol formado con todos los elementos de la figura. Los elementos que dependen de otros elementos pueden ser revelados al hacer click sobre el símbolo '+'. Con ello, se mostrarán los elementos que dieron origen (v.g. los que dieron origen a una línea pueden ser dos puntos; aunque pueden ser otros originadores. También se les llama elementos antecedentes o elementos padre).

Figura 1.3: Una figura de Dr. Geo y su descripción

1.2. Dr. Geo en la red

Dr. Geo dispone de su propio espacio en la red en el sitio OFSET en la dirección : http://www.ofset.org/drgeo
En este espacio, el usuario encontrará la siguiente información :

- la información para obtener Dr. Geo ;
- la documentación del programa;
- algunas indicaciones para involucrarse en el proyecto Dr. Geo ;
- algunas referencias para el uso pedagógico del programa.

Capítulo 2

Funciones Básicas

Este capítulo describe las herramientas usadas para construir figuras geométricas. Al final del capítulo, la configuración de las preferencias del usuario es también presentada.

2.1. Herramientas de Construcción

Estas herramientas están separadas en seis grupos disponibles desde la segunda barra de herramientas de Dr. Geo .

Figura 2.1: Categorías de herramientas de Dr. Geo y sus descripciones

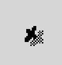
Cuando el usuario hace click sobre uno de los iconos decorados con un triangulito verde, una nueva barra vertical de iconos aparece inmediatamente. En ella se agrupan funciones de una misma familia.

De izquierda a derecha, tenemos acceso a las barras de herramientas verticales para: construir puntos, construir líneas, realizar transformaciones geométricas, calcular valores numéricos, generar macros de construcciones, y utilizar las herramientas de edición – *Otras herramientas*

Estas funciones también aparecen dentro del menú contextual de una figura, que es llamado haciendo click con el botón derecho en el fondo de la figura.

2.1.1. Herramientas de puntos

Punto Libre



Crear un punto libre en el área o un punto libre en un objeto unidimensional(segmento, semirrecta,recta, arco de círculo, círculo, lugar geométrico).

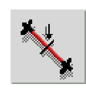
1. En el primer caso, el punto creado puede moverse a cualquier lugar del fondo de la figura; para construirlo, el usuario simplemente hace click en cualquier parte del fondo.

2. En el segundo caso, el punto está limitado a moverse en el objeto unidimensional (línea o círculo); esta pegado al objeto. Para construir este tipo de punto el usuario hace click sobre una línea (i.e. una línea recta, una semirrecta, un segmento, un círculo, un arco de círculo, etc.).

¿Cómo ubicar un punto dadas sus coordenadas ? La solución más simple consiste en colocar un punto libre y luego editar sus propiedades – herramienta Propiedades Sección 2.2.5, página 20– ajustando sus coordenadas a voluntad..

Otra posibilidad – menos flexible para este tipo de figura– consiste en colocar dos valores libres en la figura – herramienta Valor NuméricoSección 2.1.4, página 15– y luego construir el punto con coordenadas dadas por esos dos valores – herramienta Punto definido por sus coordenadas Sección 2.1.1, página 12 Cette possibilité a un avantage sur la Este procedimiento tiene una ventaja sobre el precedente: el punto construido no puede ser ya movido con el ratón; el punto está inexorablemente atado a su posición.

Punto Medio



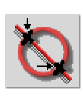
Crear el punto medio de un par de puntos o la mitad de un segmento.

1. En el primer caso, el usuario selecciona dos puntos.

2. En el segundo caso, el usuario simplemente selecciona un segmento

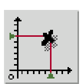
.

Intersección



Crear el(los) punto(s) de intersección de dos líneas (objetos unidimensionales, i.e. recta, semirrecta, segmento, arco de círculo, círculo o lugar geométrico). El usuario necesita seleccionar dos líneas (v.g arco de círculo y semirrecta).

Punto definido por sus coordenadas

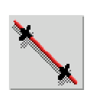


Crear un punto definido por sus coordenadas. El usuario necesita seleccionar dos números, el primer número seleccionado es la abscisa, el segundo la ordenada.

¿Cómo colocar un punto con restricciones a sus coordenadas? Esta funcion es muy utilizada cuando deseamos, por ejemplo, construir el lugar geométrico de un punto. Esta construcción supone la existencia de dos valores – ver Sección Valores Numéricos Sección 2.1.4, página 15 – el punto es construidoluego de seleccionar esos dos valores. Los valores pueden ser independientes o uno depender del otro.

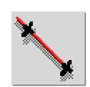
2.1.2. Herramientas de líneas

Recta



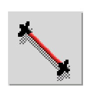
Crear una línea recta definida por sus dos puntos. El usuario selecciona dos puntos.

Semirrecta



Crea una semirrecta (también llamada rayo) definida por dos puntos. El usuario selecciona dos puntos, el primero es el origen, el segundo pertenece a la semirrecta.

Segmento



Crea un segmento definido por dos puntos.

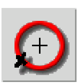
Vector



Crea un vector definido por dos puntos. El usuario selecciona dos puntos, el primer punto es el origen, el segundo es el extremo.

Una vez que el vector es creado, éste puede ser desplazado independientemente de los dos puntos que le dieron origen. Esto sigue siendo cierto para vectores construidos a través de una transformación. (Ver la sección de Transformaciones en este manual).

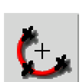
Círculo



Crea un círculo. El usuario puede crear un círculo a partir de varias opciones:

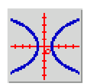
1. el centro y un punto del círculo;
2. el centro y un número (el radio del círculo);
3. el centro y un segmento cuya longitud es el radio del círculo.

Arco de Círculo



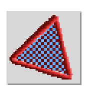
Crea un arco de círculo definido por tres puntos. El primer punto seleccionado es el origen del arco, el tercero es su extremo, el segundo es un punto sobre el arco. Los tres puntos seleccionados formarán parte del arco.

Lugar geométrico



Crea un lugar geométrico (locus en latín) dados dos puntos. El usuario selecciona dos puntos; uno es un punto libre sobre una línea (linea o círculo); el otro es un punto que depende del primero (i.e. cuando uno mueve el primero, el segundo también tiene que moverse).

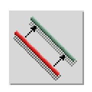
Polígono



Crea un polígono definido por n puntos. El usuario selecciona n+1 puntos delimitando el polígono. El primero y el último seleccionado deben ser los mismos, esto indica a Dr. Geo que la selección esta terminada. El objeto polígono no es un objeto como los objetos unidimensionales, no es posible colocar un punto sobre él o calcular la intersección entre un polígono y otro objeto unidimensional. Por otro lado, es posible aplicar a un polígono una transformación geométrica (rotación, reflexión, homotecia, etc.)

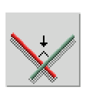
2.1.3. Herramientas de Transformación

Línea Paralela



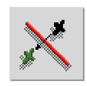
Crea una línea paralela a una dirección y que pase a través de un punto. El usuario selecciona un punto y una dirección (i.e. una línea recta, una semirrecta, un segmento o un vector).

Línea Perpendicular



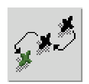
Crea una línea perpendicular a una dirección y que pasa a través de un punto. El usuario selecciona un punto y una dirección (i.e. una línea recta, una semirrecta, un segmento o un vector). (Ortogonal es un sinónimo de Perpendicular)

Simetría Axial (Reflexión)



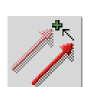
Crea la imagen de un objeto a través de una simetría axial (reflexión en una recta). El usuario selecciona el objeto a transformar y el eje de simetría (que debe ser una recta). Cuando el usuario quiere construir la imagen de una línea recta, la primera línea recta seleccionada por el usuario será la recta a reflejar.

Simetría Central



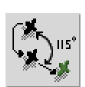
Crea la imagen de un objeto a través de una simetría central. El usuario selecciona los objetos que serán transformados y el centro de simetría (un punto). Cuando el usuario quiere construir la imagen de un punto, el primer punto seleccionado es el punto a transformar.(La simetría central es equivalente a una rotación de 180 grados).

Traslación



Crear la imagen de un objeto a través de una traslación. El usuario selecciona el objeto que será transformado y el vector de traslación. Cuando el usuario quiere construir la imagen de un vector, el primer vector seleccionado es el vector que será trasladado.

Rotación

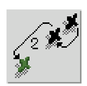


Crea la imagen de un objeto a través de una rotación. El usuario selecciona el punto que será rotado, el centro y el ángulo de la rotación. Cuando el usuario quiere crear la imagen de un punto, el primer punto seleccionado es el punto a transformar.

El ángulo puede ser seleccionado a partir de varios tipos de valores :

- **valor numérico** : el ángulo es expresado en radianes. Ejemplos de valores numéricos : valor libre, un valor devuelto por un script Guile de Dr. Geo, etc. ;
- **la medida de un ángulo geométrico formado por tres puntos** : su medida es entonces expresada en grados. Atención, pues en este caso la medida estará en el intervalo [0 ; 180] ;
- **la medida de un ángulo orientado entre dos vectores** : esta medida es expresada en grados y cubre el intervalo]-180 ; 180].

Escala (Homotecia)



Crea la imagen de un objeto a través de una transformación de escala (es decir, homotecia). El usuario selecciona el objeto a transformar, el centro de homotecia, y el factor (i.e. un número). Cuando el usuario quiera crear la imagen de un punto, el primer punto seleccionado es el punto a transformar. (Por ejemplo, si se desea hacer un polígono un tercio de grande, se selecciona un punto –centro de homotecia–, el polígono, y el valor 0.33333)

2.1.4. Herramientas Numéricas

Distancias, longitudes & Números

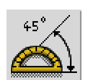


Crea un valor numérico. El valor numérico puede ser computado o editado por el usuario dependiendo de las selecciones que haga el usuario:

1. dos puntos: la distancia entre dos puntos;
2. un segmento: la longitud de este segmento;
3. un vector; la magnitud de ese vector (también llamada norma del vector);
4. un círculo: el perímetro del círculo;
5. un arco de círculo; la longitud del arco;
6. una línea recta: la pendiente de la recta;
7. una línea recta y un punto: la distancia entre la línea y el punto
8. un click sobre el fondo permite al usuario entrar un valor nuevo (i.e. un valor numérico libre)

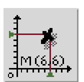
Esta última posibilidad es muy intenresante en ciertas situaciones. Por ejemplo, ella permite fijar una longitud, el radio de un círculo, la medida de un ángulo (en radianes) o las coordenadas de un punto. Estos valores numéricos pueden ser enseguida utilizados por las herramientas de construcción de círculos, de rotación de objetos o para la creación de puntos dadas sus coordenadas.

Ángulo



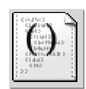
Calcula la magnitud de un ángulo definido por tres puntos o dos vectores. En el primer caso, el ángulo se considera no orientado (i.e. un ángulo geométrico con valores en el intervalo [0 ; 180°]. En el segundo caso, el ángulo se considera orientado y toma valores en el rango]-180 ; 180].

Coordenadas



Al selecciona un punto o un vector, nos da como resultado las coordenadas del punto o nos da las coordenadas del vector (coordenadas del punto final menos coordenadas del punto inicial). Esta herramienta crea tanto a la abscisa como a la ordenada.

Script Guile de Dr. Geo



Crea un script Scheme de Dr. Geo(i.e. un script en el lenguaje Scheme). El script recibe una entrada de n objetos. Siempre dará como resultado un número, que será presentado en la figura. Un script puede ser usado para obtener efectos colaterales o por el valor que da como resultado. Los Scripts Scheme de Dr. Geo están cubiertos en detalle en el Capítulo Construcciones Avanzadas Capítulo 3 y exactamente en la Sección ScriptSección 3.2, página 29.

2.1.5. Herramientas para Macros de construcciones

Crear una Macro



Extrae una secuencia de construcción dada en una figura y la convierte en la macro de una construcción.

Ejecutar una Macro



Ejecuta (i.e. "lanza" o "corre") una macro previamente construida. La macro puede haber sido recién creada o puede ser cargada desde un archivo.

(!) Las construcciones de Macro están expuestas en laSección Construcción de Macros 3.1, página 23.

2.2. Otras herramientas

2.2.1. Árbol lógico de construcción

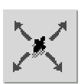
Cada figura esta asociada a un árbol lógico de construcción. Este árbol es cronológico; es decir que recuerda, desde la parte alta hasta la parte baja, el orden de construcción de la figura. Ciertas entradas del árbol pueden ser desplegadas para que aparezcan los antecedentes –es decir, los objetos originadores (objetos padres)– utilizados en la definición del objeto.

Inicialmente el árbol esta oculto (enmascarado); de hecho está replegado en el borde izquierdo de la ventana. Para hacerlo aparecer hace falta desplegarlo con la ayuda del ratón. Mover el puntero del ratón hacia el borde izquierdo de la ventana de Dr. Geo cuando éste se transforme en "¡-¿", presionar el botón del ratón y manteniéndolo presionado mueva el ratón a la derecha.

2.2.2. Mover la figura

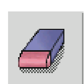
La figura puede ser movida presionando al mismo tiempo la tecla Control y el botón izquierdo del ratón al moverlo hacia donde se desea mover la figura.

Mover un Objeto



Un objeto puede ser movido al jalarlo con el ratón. La figura es redibujada con respecto a las nuevas posiciones. Casi cualquier tipo de objeto geométrico puede ser movido. Cuando es necesario, Dr. Geo mueve los puntos libres asociados con la figura. Por ejemplo, cuando el usuario mueve una línea definida por dos puntos libres, Dr. Geo moverá los dos puntos simultáneamente..

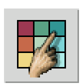
2.2.3. Borrar un objeto



Los objetos en la figura pueden ser borrados cuando se activa este menú. En cualquier momento posterior, el usuario puede anular el borrado usando la función "deshacer" de la

bara de íconos o del menú editar-¿deshacer. Inicialmente el número de anulaciones posibles es de 10, pero el usuario puede ajustar este valor a través del diálogo de preferencias (Editar -¿preferencias).

2.2.4. Editar la Apariencia de un Objeto



Cada objeto geométrico posee atributos de apariencia como son el color, el grosor, el nombre, el tamaño o la forma. Además, es posible ocultar temporalmente un objeto sin borrarlo. Por ejemplo, puede ser útil ocultar construcciones intermedias sin borrarlas. Todos estos atributos puedes ser ajustados desde un diálogo activado cuando el usuario selecciona un objeto en la figura.

El diálogo de apariencia para puntos se relaciona con cualquier tipo de objeto de punto. Desde él es posible ajustar el color, la forma, el tamaño, nombre y visibilidad.

Figura 2.2: El diálogo de estilo para un objeto tipo punto



El diálogo de apariencia para líneas se refiere a líneas rectas, semirrectas (rayos), segmentos, vectores, círculos, arcos de círculos, lugares geométricos. Desde él, es posible ajustar el color, el grosor, el nombre y su visibilidad. Cuando una recta, una semirrecta, un vector o un segmento son definidos por dos puntos que tienen nombres no vacíos, entonces el nombre de la recta es automáticamente deducido a partir de los nombres de los puntos que la originan. En este caso el usuario no puede renombrar a la recta.

El diálogo de estilo para los objetos tipo numérico y tipo polígono se refiere a todo tipo de valores (editados por el usuario, calculados por un Script Scheme de Dr. Geo o valores que representan una medida de un objeto geométrico) y de formas de polígono. (Notar que dos polígonos pueden tener colores distintos; podemos cambiarles el color)

Figura 2.3: El diálogo de apariencia para objetos tipo línea

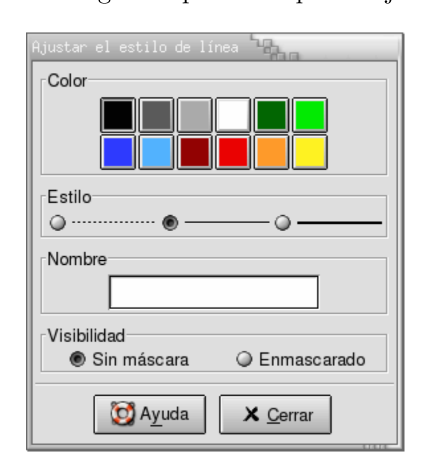


Figura 2.4: El diálogo de estilo para objetos numéricos y tipo polígono.



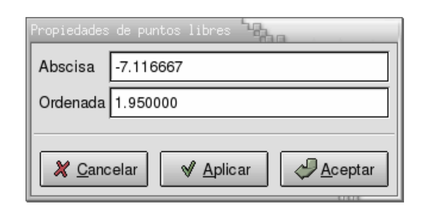
2.2.5. Editando Propiedades de Objetos



Ciertas propiedades de los objetos son ajustables por el usuario. Cuando el usuario hace click en tales objetos, un diálogo apropiado aparece. De hecho, los siguientes objetos poseen propiedades ajustables por el usuario:

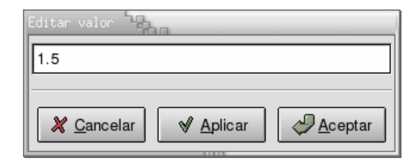
1. punto libre : la abscisa y la ordenada pueden ser editadas;

Figura 2.5: Editar las coordenadas de un punto libre



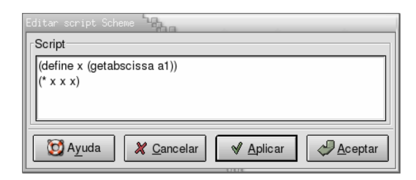
2. valor libre : su valor puede ser editado

Figura 2.6: Editar valor libre



3. script : su código puede ser editado (El código esta en el lenguaje Scheme, que es un dialecto del lenguaje Lisp)

Figura 2.7: Editando un script



2.2.6. Activar un enrejado

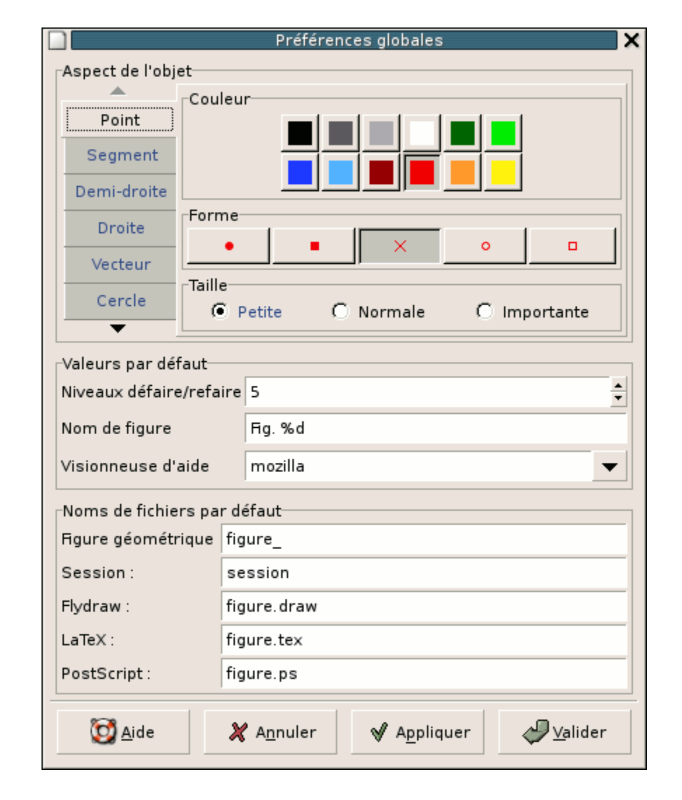
Es posible mostrar un enrejado unitario dentro de cualquier figura de Dr. Geo. El comando es accesible desde el menú `Editar->Mostrar u ocultar rejilla`. Puede ser también activado con la tecla `Ctrl-G`. Si el comando es reactivado, el enrejado es ocultado. El enrejado es unitario: cada subdivisión representa una unidad. Por último, si el enrejado esta activado al momento de guardar la figura; éste también será guardado junto con ella.

2.3. Preferencias de Usuario

2.3.1. Comportamiento Predeterminado

El comportamiento predeterminado de Dr. Geo puede ser modificado de varias maneras. Para ajustar las preferencias, el usuario va al elemento del menú `Editar->Preferencias...` para abrir el diálogo de preferencias.

Figura 2.8: Preferencias de las figuras geométricas



El diálogo se compone de dos partes ::

1. La primera parte trata sobre propiedades de las figuras geométricas. Las distintas opciones permiten al usuario dar las reglas predeterminadas para cada tipo de objeto (geométrico o numérico). Estas reglas se refieren a la apariencia de los objetos.

2. La segunda parte trata sobre preferencias globales :

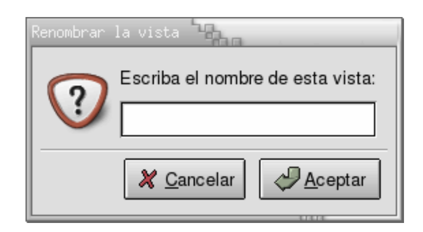
   - El nivel para Deshacer/Rehacer;
   - El nombre de la figura predeterminado que se usará cuando una nueva figura es creada. El %d es remplazado por un valor entero que es manejado por Dr. Geo, este valor es incrementado con cada nueva figura creada;

- El navegador de internet predeterminado para visualisar la ayuda en línea ;
- Los nombres predeterminadas para ser utilizados cuando se guarde una figura o una sesión ;
- Los nombres predeterminados para exportar en los formatos LaTeX y PostScript ;

2.3.2. Otras Preferencias

Además del comportamiento predeterminada de Dr. Geo, el usuario puede cambiar el nombre de una figura desde el elemento menú `Editar->Renombrar`.

Figura 2.9: Renombrar la vista de una figura



Capítulo 3

Características Avanzadas

En este capítulo presentamos características usadas para extender las características de Dr. Geo o adaptarlo a una situación pedagógica dada.

La primera es la macro de una construcción geométrica. Ella permite la extracción de la lógica de una construcción en un registro. Ya hecho esto, este registro se puede repetir o guardar en un archivo que lleve la extensión **.mgeo**, y puede ser abierto cuando sea necesario..

Los Scripts Scheme de Dr. Geo – Dr. Geo Script en inglés; abreviado DGS -representan otra forma de extender Dr. Geo. Estos scripts son en realidad elementos de figuras como cualquier otro elemento geométrico. Ellos reciben como entrada las referencias a los elementos de figura seleccionados por el usuario y dan como resultado un valor numérico, que se inserta en la figura . Son, en efecto, funciones [1] injertadas en una figura, y son evaluadas en cada actualización de la figura (i.e. cada vez que el programa requiere volver a dibujar la figura)

Los Scripts Guile de Dr. Geo pueden ser útiles por el valor que dan como resultado o por sus efectos colaterales, dependiendo de lo que el usuario desee hacer.

Extendiendo la funcionalidad de los Scripts de DrGeo, Dr. Geo puede ir todavía más allá, con la Figura Scheme de Dr. Geo: Ésta consiste en un archivo de texto que contiene una figura geométrica completamente escrita en el lenguaje de programación Scheme. La potencia de este modo de crear una figura radica en permitir una construcción usando el tipo de programación llamado funcional [2], y no solamente declarativo, como es el caso de la interfaz gráfica. De este modo, Dr. Geo se convierte en un programa que puede ser utilizado para la introducción a la programación.

Por último, la adaptación de la interfaz de usuario de Dr. Geo permite al maestro preparar una sesión de trabajo con varios documentos en los que ciertas funciones han sido bloqueadas a través de una contraseña (password). El objetivo de esto es permitir el uso del programa en ciertas situaciones pedagógicas dadas.

3.1. Macros de Construcciones

Una Macro para una una construcción es una especie de procedimiento que recibe como entrada a elementos de figuras y devuelve uno o más elementos de figuras, construidas por la macro. Ellas son construidas en un modelo definido por el usuario. Esto significa que ella (o él) tiene que elaborar la serie de pasos para la construcción sólo una vez, a partir de entonces ordena a Dr. Geo grabar esta serie de pasos en una macro (macro-construction). Esta macro puede ser entonces guardada en un archivo con extensión **.mgeo**.

Para grabar una serie de pasos para la construcción, Dr. Geo necesita saber cuáles son los elementos iniciales de la serie y los elementos que se van a dar como resultado.

[1] O procedimientos para los que se iniciaron en el lenguaje Pascal.
[2] Por ejemplo, permite la recursión.

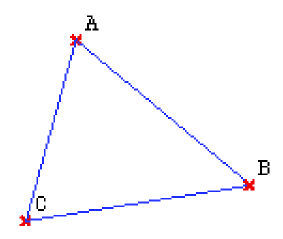
Desde luego que los elementos que se den como resultado deben depender *solamente* de los elementos iniciales, [3], de otro modo Dr. Geo no podría deducir los elementos producidos a partir de los dados.

De este modo, Dr. Geo deduce la lógica de la serie de pasos de la construcción y la guarde en una macro. El usuario puede ejecutar esta macro con sólo especificar los elementos de entrada (que deben coincidir con los parámetros que necesita la macro) en la figura. Entonces la macro crealos elementos especificados como resultado.

(!) Los elementos invisibles de las figuras intermedias también son construidos por la macro. Estos elementos son necesarios para crear los elementos resultantes

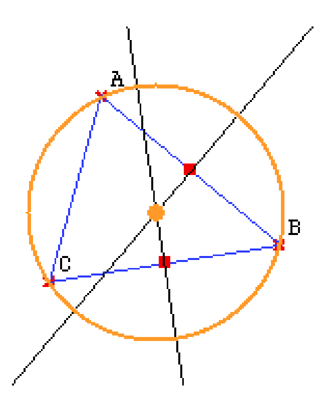
Para ilustrar la característica de macros para construcciones, supondremos que el usuario desea grabar la construcción de un círculo que pasa por tres puntos. Supondremos, además, que se desea obtener el centro del círculo.

Figura 3.1: Nuestra figura inicial



Antes de la creación de la macro, el usuario necesita construir la figura final, ésta es usada como un molde con el cual se crea la macro.

Figura 3.2: Nuestra figura con la construcción resultante



3.1.1. Crear una Macro

En esta etapa, la serie de pasos para la construcción ya está hecha. Ahora el usuario necesita ordenar a Dr. Geo que quiere una macro a partir de esta serie de pasos. Puede

[3]Esta restricción ha sido posteriormente flexibilizada, lo cual permite ir más allá con las macors. Ver Crear un polígono regular Sección 6.1, página 69.

llamar la función `Construir una macro` de la barra de íconos o del menú contextual que aparece al pulsar el botón derecho sobre el fondo de la figura..

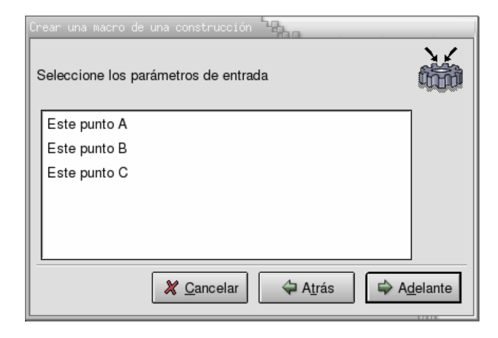
Desde el menú de creación de la macro, el usuario selecciona los parámetros de entrada y de salida, el nombre y la descripción de la macro.

Figura 3.3: La primera parte del menú de creación de la macro



La segunda parte del diálogo sirve para seleccionar los parámetros de entrada. En nuestro ejemplo, estos son los tres puntos iniciales. El usuario sólo necesita llegar a este lugar y seleccionar los tres puntos en la figura. Los elementos seleccionados parpadearán.

Figura 3.4: Segunda parte, los tres puntos ya están seleccionados



En la tercera parte del diálogo, el usuario selecciona los parámetros de salida. En nuestro ejemplo, deseamos que el círculo y el centro sean los parámetros de salida para la macro. El usuario procede entonces como en el caso de los parámetros de entrada par seleccionarlos.

En la cuarta parte del diálogo, el usuario escribe el nombre y la descripción de la macro. Estos datos son desplegados cuando el usuario ejecuta una macro, de modo que ayuda a distinguir una macro de otra.

En la última parte del diálogo (la quinta), el usuario crea la macro haciendo click sobre el botón `Aplicar`(finalizar la construcción). Alternativamente, el usuario puede regresar a las partes previas para ajustar los parámetros de la macro.

(!) Si los parámetros de entrada y de salida no coinciden (Dr. Geo no puede extraer la lógica de la construcción), la macro no puede ser construida. En este caso, el usuario necesita reconsiderar la selección de los parámetros de entrada y de salida. Puede regresar a la segunda o a la tercera parte del menú de creación para ajustar sus elecciones.

Figura 3.5: La tercera página, el círculo y su centro ya están seleccionados

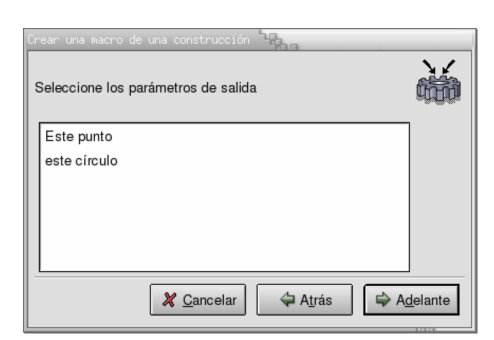
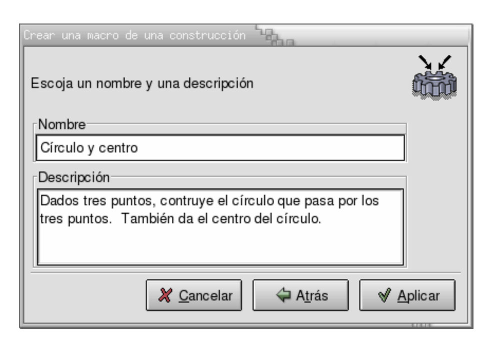


Figura 3.6: La cuarta parte, el nombre y la descripción de la macro



En este momento la macro está construida y grabada en Dr. Geo. En la siguiente sección, veremos cómo usarla.

3.1.2. Ejecutar una Macro

Para ejecutar la macro de una construcción, el usuario llama la función `Ejecutar macro previamente construida` de la barra de íconos o desde el menú contextual que se obtiene pulsando el boton derecho. Una ventana de diálogo que describe el procedimiento se inserta entonces.

Desde la ventana de diálogo, el usuario selecciona la macro. En la segunda parte, selecciona la macro de la lista en la parte más alta del diálogo. Una vez que la macro fue seleccionada, puede hacer click directamente en los parámetros de entrada en la figura. Tan pronto como todos los parámetros de entrada sean seleccionados, la macro es ejecutada y los elementos de salida aparecen.

En nuestro ejemplo, la macro necesita tres parámetros de entrada (tres puntos) y crea a partir de ellos un círculo y un punto. Para ejecutar nuestra macro, necesitamos una figura con al menos tres puntos.

Una vez que nuestra macro es ejecutada usando esos tres puntos, obtenemos el círculo deseado y también su centro.

Con la ayuda del menú Macros

Existe otro procedimiento –más rápido– para ejecutar una macro. La barra de menú principal de Dr. Geo contiene un menú `Macro-constructions`. Este menú esta poblado con

Figura 3.7: El usuario selecciona los parámetros de entrada directamente en la figura

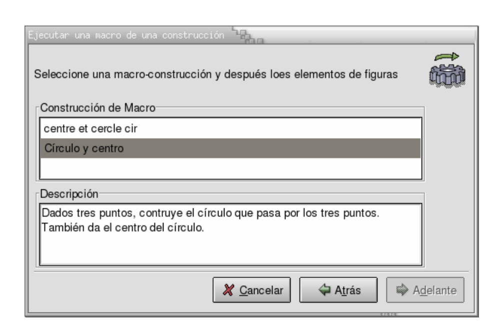


Figura 3.8: Una figura con tres puntos

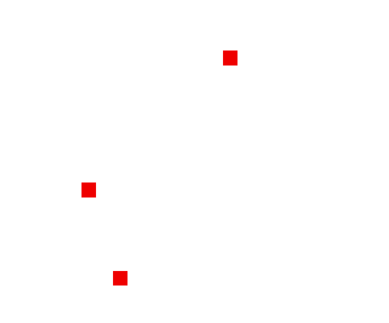
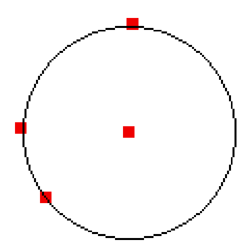


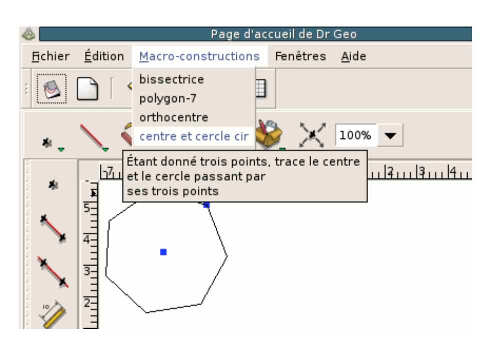
Figura 3.9: La figura resultante con el círculo y su centro



los nombres de las macros cargadas en la memoria del programa. Para ejecutar una macro, el usuario selecciona directamente la macro de su preferencia.

En outre un passage du pointeur souris au dessus de chaque item de menu fait apparaître une info-bulle de la description de la macro. L'utilisateur peut ainsi avoir rapidement une explication sur l'ensemble des macro-constructions.

Figura 3.10: Ejecutar una macro directamente desde el menú `Macro-constructions`



3.2. Script Scheme de Dr. Geo

Dr. Geo es compatible con Guile. Esto quiere decir que es posible ejecutar un script del lenguaje Scheme en Dr. Geo. Pero ¿qué es Guile? Leemos en el manual de Guile :

> *Guile es un intérprete del lenguaje de programación Scheme, concebida para su uso en una gran variedad de ambientes*

Las siguientes citas describen de modo preciso como es Guile usado dentro de Dr. Geo:

> *Como un shell, Guile puede correr interactivamente, leyendo expresiones del usuario, evaluándolas, y mostrando los resultados, o como un intérprete para un script, leyendo y ejecutando código de Scheme a partir de un archivo. Por otro lado, Guile también está disponible como una biblioteca (library) de objetos, lo que permite a otras aplicaciones fácilmente incorporar un intérprete de Scheme completo. Un programa puede usar Guile como un lenguaje de extensión, un lenguaje de configuración simple y poderoso, o como un "pegamento" multipropósito que conecta primitivas elaboradas dentro del programa.*

En Dr. Geo, una API (Interfaz de programa para una aplicación, –Application Program Interface– en Inglés) está disponible a partir del intérprete Guile. Esta API es un conjunto de "enlaces" en la maquinaria geométrica. Por lo tanto el usuario puede escribir un script para manipular elementos de figuras (geométricas y numéricas). Asimismo, ya que los scripts son figuras como cualquier otra, los scripts no necesitan ser guardados en un archivo separado; son guardados en el archivo de la figura. En lo que sigue usaremos el acrónimo DGS para referirnos a un Script Scheme de Dr. Geo ("Dr. Geo Script").

3.2.1. DGS a través de ejemplos

La herramienta para crear un DGS esta disponibe al seleccionar la sección Valor Numérico dentro del menú contextual que se obtiene al hacer click con el botón derecho del ratón en el fondo de la figura. Un DGS puede recibir de 0 a n parámetros de entrada.

Después de haber escogido el DGS, basta hacer click en los objetos geométricos (o numéricos) que constituirán los parámetros de entrada. Al terminar de escoger los parámetros de entrada, se debe hacer click en un área libre del fondo de la figura; con ello, el script sera visible.

En lo que sigue nos proponemos trabajar con algunos ejemplos de DGS que permitirán comprender fácilmente su uso y su potencialidad.

Los DGS, como las macros, dan una dimensión particular a Dr. Geo, Le permiten, junto con las macros[4], llevar a cabo cosas que los autores del programa no incorporaron o no quisieron incorporar: "Ir hacia donde ellos no fueron o no quisieron ir".

Es también importante comprender que la mayor parte de las funciones del intérprete GNU Guile están disponibles a través del DGS. Esto es particularmente cierto para las librerías de funciones[5] que utilizaremos ampliamente.

DGS sin parámetros de entrada El procedimiento para crear un script sin parámetros de entrada es el siguiente :

1. Después de haber escogido Script Guile de Dr. Geo Sección 2.1.4, página 16, hacer click en el lugar de la figura donde se desee colocar el script. Como deseamos que el script no tenga parámetros de entrada, debemos prestar atención y no hacer click

[4]Las macros constituyen el aspecto geométrico, en tanto que los DGS abarcan también el aspecto numérico; pero no sólo eso, pues nos permiten utilizarlos con un espíritu de programador ("hacker", en inglés, sería un término más apropiado)

[5]En particular, las funciones matemáticas

erróneamente en algún objeto geométrico; no sea que Dr. Geo considere ese objeto como un parámetro de entrada [6].

2. Una vez que el DGS ha sido creado, aparece en la figura la cadena de caracteres “Dr. Genius”. Todo script recién creado contiene un comando predeterminado que muestra ese mensaje. Usted puede editarlo seleccionando propiedades del objeto Sección 2.2.5, página 20.

3. Una vez que esta herramienta ha sido seleccionada, hacer click sobre el script –o para ser más precisos sobre su valor– de su elección. Una ventana se abrirá con el contenido del script y le permitirá editarlo. En lo que sigue, usaremos ese diálogo para escribir nuestros ejemplos.

Un generador de números aleatorios y otros ejemplos Si usted desea un generador de números aleatorios, nada es más simple: simplemente su script contiene únicamente la siguiente línea (recuerde hacer click en aplicar y luego cerrar...):

```
(random 10)
```

Así, cada vez que se dibuja la figura, este script devuelve un número entero aleatorio en el intervalo [0 ; 10[(es decir, [0 ; 10))

Si usted prefiere un número real en el intervalo [0 , 1), utilice el siguiente script:

```
(random:uniform)
```

(!) Algunas precisiones :

- El valor devuelto por el script es el valor calculado por la última línea del script. En nuestros ejemplos, se trata del valor devuelto por la función que usamos.
- La última línea debe devolver un número real; de otro modo Dr. Geo imprimirá “Resultado no imprimible”.
- Si lo que se desea es mostrar el valor de una variable, basta poner esa variable en la última línea.

Cálculo de algunas constantes comunes Para calcular el valor aproximado de π :

```
(acos -1)
```

O de e :

```
(exp 1)
```

Los valores devueltos por estos DGS pueden ser inmediatamente utilizados como todos los demás valores numéricos que Dr. Geo genera. Por todo lo anterior, los DGS son verdaderamente nuestros aliados. Pero eso no es todo, los DGS pueden hacer muchas más cosas interesantes cuando reciben parámetros de entrada. Enseguida veremos cómo.

[6] Si por accidente hacemos click sobre un objeto, el seleccionar de nuevo la herramienta Script en el menú bastará para anular nuestro error

DGS con al menos un parámetro de entrada El procedimiento para crear un DGS con un parámetro de entrada es esencialmente el mismo. Justo después de haber seleccionado la herramienta script, basta hacer click sobre el objeto que será el parámetro de entrada y enseguida hacer click sobre el fondo de la figura, en el lugar donde se desee tener al nuevo script.

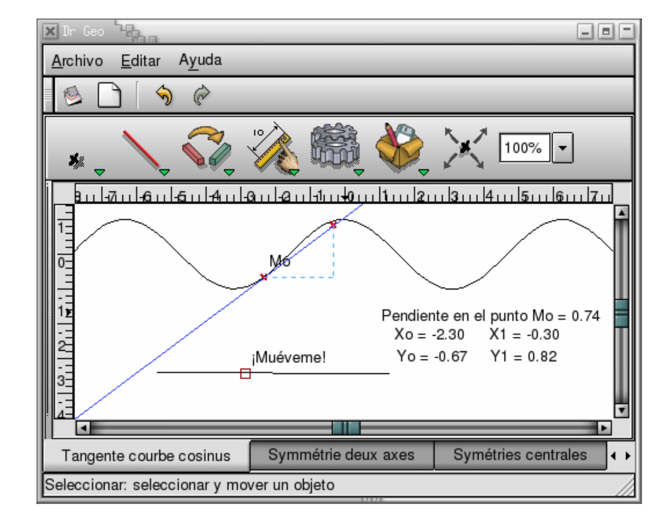
Enseguida, dentro del script, el parametro de entrada será referido por la variable $a1$. En caso de tengamos dos o más parámetros de entrada, utilizaremos, respetando el orden en que fueron seleccionados, las variables $a1, a2, a3, a4$, etc. dependiendo del número de objetos seleccionados.

Dependiendo del objeto que seleccionemos, diversos métodos estarán disponibles para obtener valores, como pueden ser coordenadas, longitud, etc. Estos métodos son mostrados en la Sección Métodos de referencia para un DGS 3.2.2, página 33.

En lo que sigue, expondremos gradualmente un ejemplo no muy complicado en que construiremos la curva de una función así como la tangente en un punto de la curva que podremos mover.

La figura final es mostrada por Dr. Geo. Esta se llama slope.fgeo(/usr/share/drgeo/examples/figures/slope.fgeo)

Figura 3.11: La figura que obtendremos



Definir un valor en un intervalo dado. En una nueva figura, empezamos colocando dos puntos y el segmento que ellos determinan. Sobre este segmento colocamos un punto libre al que llamaremos "¡Muéveme!"("Move me!" en inglés). Este punto nos sirve para determinar un valor numérico que identifica la posición del punto dentro del segmento; esto lo lograremos dentro del script.

En seguida crearemos un script que tiene como único parámetro de entrada el punto "¡Muéveme!".

Ya que $a1$ es la variable en el script del punto "¡Muéveme!", el script siguiente (que llamaremos Xo) nos dará un valor decimal comprendido entre [-10 ; 10] .

```
(define x (getAbscissa a1))
(* 20  (- x 0.5))
```

Cabe aclarar algunas cosas. En la primera línea, `(getAbscissa a1)` permite obtener la "abscisa curvilínea" del objeto al que se refiere $a1$[7] . En la segunda línea calibramos el valor.

[7]En este punto x toma un valor entre 0 y 1 inclusive. La "abscisa curvilínea" del objeto es 0 en un extremo, y va creciendo hasta ser 1 en el otro extremo; es 0.5 en la mitad, 0.25 a un cuarto, etc.

En este caso la expresión significa $20 \times (x - 0{,}5)$. Dado que ésta es la última línea del script, el resultado de esta última operación será el valor que el script devuelva, y será un número decimal en el intervalo $[-10; +10]$ como se deseaba[8].

Finalmente, llamemos este script Xo. (Esto se puede hacer en el menú contextual: seleccione otros, luego apariencia, y haga click en el script para ponerle nombre).

Dibujar la gráfica de una función. El valor obtenido por el script precedente nos servirá, a través de otro script, para calcular la imagen del punto bajo la función $x \to cos(x)$. Este segundo script tiene como parámetro de entrada al script Xo.

```
(define x (getValue a1))
(cos x)
```

Subrayemos que en `(getValue a1)` el objeto al que se refiere $a1$ no es el punto "¡Muéveme!", sino el script Xo, y que no estamos obteniendo la "abscisa curvilínea" de un punto, sino el valor de un número (en este caso del script Xo, que dentro del segundo script es llamado $a1$). Llamamos a este script Yo

Por ejemplo, si el punto "¡Muéveme!" estaba a la mitad entonces su "abscisa curvilínea" es 0.5, de modo que Xo toma el valor $20 \times (0{,}5 - 0{,}5) = 0$, y Yo toma el valor cos(0)=1.

Enseguida crearemos el punto Mo de coordenadas (Xo ; Yo). Es un punto de la curva de la función $x \to cos(x)$. Para dibujar la porción de la gráfica con dominio [-10 ; 10], crearemos el lugar geométrico del punto Mo cuando el punto "¡Muéveme" describe el segmento. Con ello obtendremos la gráfica!(Recordar que para crear Mo, hay que seleccionar, en el menú contextual, punto, coordenadas y luego hacer click en los scripts que dan las coordenadas. Para lograr el lugar geométrico, recordar seleccionar en el menú contextual curva, lugar geométrico y luego seleccionar los puntos "¡Muéveme!" y Mo.)

Calcular y dibujar la tangente a la gráfica en un punto Para dibujar la tangente en Mo=(Xo,Yo), hace falta tener la pendiente en ese punto. Usando Cálculo Diferencial, sabemos que la pendiente esta dada por la derivada, que en este caso es $x \to -sin(x)$ evaluada en el punto Xo. De modo que crearemos un script que tiene como parámetro de entrada al script Xo :

```
(- 0 (sin (getValue a1)))
```

La notación prefija (i.e "* 2 3" en vez de "2 * 3") utilizada por Scheme/Guile puede parecer poco intuitiva, pero es cuestión de simplemente acostumbrarse a ella; es simplemente decir "El producto de dos y tres" en lugar de "dos por tres".

Llamemos al script recién creado "Pendiente en el punto Mo".

Resta, pues dibujar la tangente. Para esto calcularemos, primero, las coordenadas de un segundo punto – M1– de esa recta.

Comencemos por su abscisa (No confundir con la "abscisa geométrica"); por ejemplo: $X_1 = X_0 + 2$. Para hacer esto crearemos un script que tendrá como parámetro de entrada al script Xo , y al que llamaremos X1:

```
(define x1 (getValue a1))
(+ x1 2)
```

Encontremos en este momento la ordenada del punto M1. Para ello necesitamos:

- Mo (referencia $a1$) ;
- la pendiente en Mo ($a2$) ;
- la abscisa X1 ($a3$).(No confundir con "la abscisa geométrica")

[8]Ya que x estaba entre 0 y 1, $20 \times (x - 0{,}5)$ estará entre -10 y 10.

Después de seleccionar como parámetros de entrada al punto Mo, a la pendiente en Mo(que llamamos “Pendiente en el punto Mo”, “Slope at Mo” en inglés), y a X1, determinamos la ordenada de M1, que llamaremos Y1, a través del siguiente cálculo:

$Yo + m \times (X1 - X0)$:

```
(define x0 (car (getCoordinates a1)))
(define y0 (cadr (getCoordinates a1)))
(define m (getValue a2))
(define x1 (getValue a3))
(+ (* m (- x1 x0)) y0)
```

En relación con la función `(getCoordinates a1)` en la que $a1$ debe ser una referencia a un objeto de tipo punto, el método devuelve una lista que contiene las coordenadas del punto, en este caso Mo. La instrucción `car` permite extraer el primer elemento de esta lista; la instrucción `cadr`, el segundo. El resto del script debe resultar claro.

Habiendo nombrado este último script Y1, construimos el punto M1=(X1 ; Y1), y, para terminar, construimos la tangente (MoM1). (Para esto ya no necesitamos scripts; podemos usar el menú, escogiendo punto dadas sus coordenadas y la recta dados dos puntos).

Por supuesto, habría sido posible utilizar unos dos o tres scripts en lugar de la gran cantidad que usamos. Sin embargo, esperamos que estos pequeños ejemplos logren en usted el deseo de experimentar por usted mismo los DGS.

3.2.2. Métodos de referencia para los scripts de Dr. Geo

Las secciones siguientes contienen la descripción de los métodos disponibles para los DGS. Estos métodos están clasificados de acuerdo al tipo de objeto geométrico o numérico.

Punto

`valor (getAbscissa punto)`

`punto`: Referencia a un punto sobre una curva
Devuelve: La “abscisa geométrica” del punto sobre la curva.El valor pertenece al intervalo [0 ; 1]
Ejemplo:

```
(define x (getAbscissa a1))
(* x 10)
```

`(setAbscissa punto x)`

`punto`: Referencia a un punto libre sobre una línea
`x`: Valor decimal en el intervalo [0 ; 1] que representa a la nueva “abscisa geométrica”
Ejemplo:

```
(setAbscissa a1 0.5)
```

`lista (getCoordinates punto|vector)`

Devuelve las coordenadas de un punto o de un vector.
`punto/vector`: Referencia a un punto o un vector
Devuelve: Lista que contiene las coordenadas del punto o del vector
Ejemplo:

```
(define c (getCoordinates a1))
(define x (car c))
(define y (cadr c))
(+ (* x x) (* y y))
```

`(setCoordinates punto coord)`

Asigna coordenadas a un punto
punto: Referencia a un punto libre en el plano
coord: Lista de dos números decimales
Ejemplo:
```
(define l (list 1.4 (random 5)))
(setCoordinate a1 l)
```

Recta, Semirrecta, Segmento, Vector

`valor (getSlope dirección)`

dirección: Referencia a un objeto de tipo recta, semirrecta, segmento o vector
Devuelve: La pendiente en esa dirección
Ejemplo:
```
(define p (getSlope a1))
```

`valor (getUnit dirección)`

dirección: Referencia a un objeto de tipo recta, semirrecta, segmento o vector
Devuelve: Una lista que contiene las coordenadas de un vector unitario en la dirección dada
Ejemplo:
```
(define v (getUnit a1))
```

`valor (getNormal dirección)`

dirección: Referencia a un objeto de tipo recta, semirrecta segmento o vector
Devuelve: Una lista que contiene las coordenadas de un vector normal a la dirección dada
Ejemplo:
```
(define n (getNormal a1))
```

`valor (getNorm vector)`

vector: Referencia a un vector
Devuelve: La norma (i.e., magnitud, longitud) de ese vector
Ejemplo:
```
(define n (getNorm a1))
```

`valor (getLength segmento)`

segmento: Referencia a un segmento
Devuelve: La longitud de ese segmento
Ejemplo:
```
(define l (getLength a1))
```

Círculo, Arco de círculo

`lista (getCenter círculo|arco)`

círculo|arco: Referencia a un círculo o un arco de círculo
Devuelve: Lista que contiene las coordenadas del centro de un círculo o de un arco de círculo
Ejemplo:

```
(define c (getCenter a1))
(car c)
```

`valor (getRadius círculo|arco)`

círculo|arco: Referencia a un círculo o un arco de círculo
Devuelve: Radio del círculo o del arco de círculo
Ejemplo:

```
(define r (getRadius a1))
```

`valor (getLength círculo|arco)`

círculo|arco: Referencia a un círculo o un arco de círculo
Devuelve: Perímetro del círculo o longitud del arco de círculo
Ejemplo:

```
(define l (getLength a1))
```

Funciones Numéricas

`valor (getValue número)`

número: Referencia a un número
Devuelve: Valor de ese número
Ejemplo:

```
(define a (getValue a1))
(define b (getValue a2))
(+ a b)
```

`(setValue número v)`

número: Referencia a un número
v: Valor decimal
Ejemplo:

```
(define v (getValue a1))
(setValue a2 v)
```

Ángulo

`valor (getAngle ángulo)`

ángulo: Referencia a un ángulo orientado o geométrico
Devuelve: La medida de ese ángulo en grados. Para obtener la medida en radianes, utilice la función `getValue`
Ejemplo:

```
(define angle1 (getAngle a1))
(define angle2 (getAngle2))
(define angle3 (getAngle a3))
(+ angle1 angle2 angle3)
```

Otros

`(move item t)`

item: Referencia a un objeto de la figura
t: vector de dos dimensiones
Ejemplo:

```
(define v (vector .1 0))
(move a1 v)
```

3.3. Figura Scheme de Dr. Geo

Las *Figuras Scheme de* DR. GEO – (FSD) – son figuras escritas en un lenguaje relativamente natural. No se trata entonces de construir una figura con la ayuda de la interfaz gráfica de DR. GEO sino más bien de describir una figura dentro del lenguaje Scheme(Una variante de Lisp). Hemos cuidado mucho que la sintaxis utilizada sea fácil y concisa. Además, el conjunto de palabras clave utilizadas en Scheme para para describir una figura simple son adaptables de modo que aparezcan en Español, Francés, Inglés, etc. . Una combinación de varios idiomas es también posible; pero no es deseable.(N. del T. es por ello que varias palabras clave en este manual aparecen en Español. Basta con traducir un archivo en el directorio scm donde DR. GEO está instalado; notar que en la traducción hecha del archivo, se incluyó una versión de los comandos con acentos y eñes, así como otra sin acentos y sin eñes. Las palabras clave no traducidas seran escritas en inglés.)

3.3.1. Algunos Ejemplos

(N.del T. En lo que sigue se usarán las palabras "referencia" y "nombre". "referencia" es una variable que apunta a un objeto. "nombre" es el rótulo que aparece en la figura. A veces se usará el termino variable; pero "referencia" es más preciso en otros casos.)

En sí, Scheme es un lenguaje de muy alto nivel. Cuando una figura es definida en este lenguaje, disponemos también de todo su poder para, por ejemplo, definir recursivamente alguna parte de la figura o bien para colocar aleatoriamente ciertos objetos de tal suerte que, cada vez que se cargue la figura, los objetos aparecerán ligeramente distintos. En pocas palabras, los FSD no están atados a las limitantes de la interfaz gráfica, y además poseen todo el poder del lenguaje Scheme. Un FSD es entonces un archivo con extension **.scm** creado con la ayuda de un editor de texto, el cual es abierto en DR. GEO con la ayuda del comando `Archivo->Evaluar`.

Comencemos por estudiar un ejemplo sencillo de FSD :

```
(nueva-figura "Mi figura")
```

Este es el script más pequeño que podemos definir. Después de ser cargado en DR. GEO, simplemente crea una nueva figura vacía con el nombre "Mi figura". Podemos escribir comandos `(nueva-figura "Mi figura")` tantas veces como deseemos.

Veamos ahora un segundo ejemplo :

```
(nueva-figura "Mi figura")
(sea Punto "A" libre 1.2 -2)
```

Este FSD define una figura con un punto libre A de coordenadas iniciales $(1,2\ ;\ \ -2)$. Como podemos ver la sintaxis de la definición de un objeto geométrico es relativamente cómoda, a tal grado que esa expresada en nuestra lengua madre. Esto es de interés especial para la enseñanza secundaria. De hecho, todos los comandos de definición de objeto tienen una sintaxis común. Tal sintaxis es como sigue:

1. Se comienza siempre con la palabra clave `sea`, la cual indica que deseamos definir un nuevo objeto

2. Inmediatamente le sigue la clase de objeto; en este caso `Punto`.

3. El nombre del objeto va enseguida, A, debe siempre estar entre dos comillas ". Si no deseamos nombrar al objeto, de todos modos es necesario dar un nombre vacío como sigue: "".

4. Para terminar, necesitamos precisar, dentro de la clase, qué tipo de objeto, en nuestro ejemplo, la clase es punto, y el tipo es `libre`. Esto significa que el punto A es libre.

5. El tipo de objeto es seguido por una lista de argumentos que depende de la clase de objeto y de que tipo de objeto. En nuestro ejemplo, esta lista se compone de dos números, las coordenadas del punto libre A.

Prosigamos con un tercer ejemplo :

```
(define (triángulo p1 p2 p3)
  (Segmento "" extremos p1 p2)
  (Segmento "" extremos p2 p3)
  (Segmento "" extremos p1 p3))

(define (azar)
  (- 8 (* 16 (random:uniform))))

(nueva-figura "Mi figura")

(sea Punto "A" libre (azar) 0)
(sea Punto "B" libre 5 0)
(sea Punto "C" libre (azar) 5)

(triángulo A B C)
```

(N. del T. En este ejemplo, dos palabras están en inglés: “random” que significa “azar”, y “uniform” que significa “uniforme”. Como se comentó antes es posible traducirlas modificando el archivo dentro del directorio scm donde Dr. Geo esté instalado. El archivo para español es: `/usr/share/drgeo/scm/drgeo_scm_interface_constant_es.scm`).

Este ejemplo es particularmente interesante, nos muestra tres cosas importantes :

1. La definición de una construcción de alto nivel no prevista por los autores de Dr. Geo. En el programa que acabamos de ver hemos definido la función `triángulo` la cual, a partir de tres puntos construye el triángulo que pasa por esos tres puntos. Podemos comparar esto con las macros; pero con un grado de libertad mucho más amplio e importante.

2. La definición de las funciones asociadas, en este caso definimos la función `azar` que devuelve un número decimal comprendido entre -8 y 8. Utilizamos esta función para añadir un elemento de azar en ciertos puntos en nuestra figura. De este modo cada vez que se cargue la figura, ésta será ligeramente distinta.

3. En efecto, la utilización de la palabra clave `sea` no es obligatoria; nosotros la utilizamos para guardar una referencia al objeto creado. Por ejemplo, dentro de la función `triángulo` no guardamos referencias para los segmentos creados. Por otro lado, cuando definimos los puntos A, B y C necesitamos guardar una referencia para ellos, estas referencias se llamarán igual[9]; pero sin las comillas; así nuestras referencias son:

[9] Desde el punto de vista del lenguaje Scheme, estas referencias son símbolos que apuntan hacia una estructura interna del objeto –un prototipo– mientras que los nombres son simplemente cadenas de caracteres.

`A`, `B` y `C`. En lo que sigue llamaremos **símbolos** a esas variables, esta es la terminología precisa del lenguaje Scheme. De este modo, al llamar a la función `triángulo`, pasamos como parámetros a los símbolos `A`, `B` y `C` que son utilizados para definir nuestros tres segmentos.

Notar que cuando definimos los segmentos, no les dimos ningún nombre; en ese caso Dr. Geo les dará un nombre que está determinado a partir del nombre de sus extremos. Nuestros tres segmentos tendrán entonces los nombres $[AB]$, $[BC]$ y $[AC]$.

Para concluir esta sección, veamos un último ejemplo :

```
(sea Punto "A" libre 1 0)
(sea Punto "B" libre 5 0)
(sea Recta "d1" 2puntos A B)

(envía A color amarillo)
(envía A forma redonda)
(envía A tamaño grande)
(envía B oculto)
(envía d1 grosor punteado)
```

Los tres primeros comandos crean dos puntos y una recta. La parte que nos interesa particularmente es la serie de comandos `envía`. Estos comandos permiten la comunicación con un objeto cuyo símbolo hemos guardado; en este caso tenemos los símbolos `A`, `B` y `d1`. El comando consiste en enviar un mensaje a un objeto. Su primer parámetro de entrada es el objeto con el que deseamos comunicarnos, el segundo argumento es el mensaje, los restantes argumentos están determinados por tipo de mensaje que es enviado. Por ejemplo `(envía A color amarillo)` envía el mensaje `color` con parámetro de entrada `amarillo`. El punto A toma entonces el color amarillo. Es fácil comprender el sentido de los otros comandos `envía`. Estos serán explicadosen la sección siguiente.

Hemos terminado nuestra pequeña visita guiada a las *Figuras Scheme de Dr. Geo, FSD.* En las secciones siguientes expondremos el conjunto de comandos disponibles para definir las FSD.

3.3.2. Métodos de referencia para las figuras Scheme de Dr. Geo

La definición de los objetos en un documento FSD se hace a través de prototipos. Los prototipos son en cierto sentido objetos a los que podemos interrogar y modificar como veremos en lo que sigue.

Empero, antes de la definición de objeto alguno en un FSD, el FSD debe ser creado con el comando `nueva-figura` en el principio del archivo.

Comandos generales

`(nueva-figura nombre)`

`nombre`: Cadena de caracteres
Devuelve: No devuelve ningún valor. Es llamada por el efecto que produce, no por el valor que devuelve. Este comando crea una figura nueva. Los objetos creados posteriormente se crean dentro de esta figura hasta que se vuelva a llamar este comando.
Ejemplo:
`(nueva-figura "Mi 1er figura")`

Definición de objetos en una figura

Un objeto puede ser definido a través de varias sintaxis :

- `(sea Punto "p1" tipo args)` El punto es creado y su referencia guardada en la variable `p1`. Esta sintaxis utiliza una macro Scheme.
- `(Punto "Nombre" tipo args)` El punto es creado pero ninguna referencia al punto es conservada.
- `(define p1 (Punto "Nombre" tipo args))` El punto es creado y su referencia guardada en la variable `p1`.
- `(set! p1 (Punto "Nombre" tipo args))` El punto es creado y su referencia es copiada en la variable, ya existente, `p1`.

Si unos objetos son creados después del cuerpo de una función, utilice ya sea la forma `set!` o la forma especial de Scheme `let`. Es importante subrayar que la llamada se hace a una función que devuelve una referencia al objeto creado.

Para saber más sobre la correspondencia entre los nombres de los comandos Scheme en Español y en Inglés, vea el archivo `/usr/share/drgeo/scm/drgeo_scm_interface_constant_es.scm`.

Punto

`prototipo (Punto nombre libre x y)`

nombre: Cadena de caracteres que designa al nombre del objeto
x: La abscisa del punto
y: La ordenada del punto
Devuelve: Referencia a un punto libre del plano con coordenadas iniciales `x` y `y`.
Ejemplo:

```
(define p1 (Punto "A" libre 1.2 (acos -1)))
```

`prototipo (Punto nombre sobre-la-curva curva x)`

nombre: Cadena de caracteres que designa al nombre del objeto
curva: Referencia a una curva (recta, semirrecta, segmento, arco, círculo, etc..)
x: Abscisa curvilínea (la "abscisa geométrica") del punto libre; el valor pertenece al intervalo [0 ; 1]
Devuelve: Referencia a un punto libre sobre la curva
Ejemplo:

```
(Punto "M" sobre-la-curva s1 0.5)
```

`prototipo (Punto nombre mitad-2pts p1 p2)`

nombre: Cadena de caracteres que designa al nombre del objeto
p1: Referencia a un punto
p2: Referencia a un punto
Devuelve: Referencia al punto medio de los dos puntos.
Ejemplo:

```
(sea Punto "A" libre 1 1)
(sea Punto "B" libre 4 4)
(Punto "I" mitad-2pts A B)
```

`prototipo (Punto nombre mitad-segmento s)`

nombre: Cadena de caracteres que designa al nombre del objeto

s : Referencia a un segmento
Devuelve: Referencia a la mitad de un segmento.
Ejemplo:
`(Punto "L" mitad-segmento s)`

`prototipo (Punto nombre intersección l1 l2)`

nombre : Cadena de caracteres que designa al nombre del objeto
l1 : Referencia a una línea(recta o curva)
l2: Referencia a una línea(recta o curva)
Devuelve: Referencia al punto de intersección de las dos líneas
Ejemplo:
`(Punto "I" intersección recta segmento)`

`prototipo (Punto nombre intersección2 l1 l2)`

nombre : Cadena de caracteres que designa al nombre del objeto
l1 : Referencia a una línea(recta o curva)
l2: Referencia a una línea(recta o curva)
Devuelve: Referencia al segundo punto de intersección de las dos líneas cuando una de las dos líneas es del tipo arco de círculo o del tipo círculo.
Ejemplo:
`(Punto "I" intersección2 recta círculo)`

Recta

`prototipo (Recta nombre 2puntos p1 p2)`

nombre : Cadena de caracteres que designa al nombre del objeto
l1 : Referencia a un punto
l2: Referencia a un punto
Devuelve: Referencia a una recta que pasa por los dos puntos.
Ejemplo:
```
(sea Punto "A" libre 0 0)
(sea Punto "M" libre 1 2)
(Recta "" 2puntos A M)
```

`prototipo (Recta nombre paralela p d)`

nombre : Cadena de caracteres que designa al nombre del objeto
p : Referencia a un punto
d: Referencia a una dirección (recta, segmento, vector,...)
Devuelve: Referencia a una recta paralela a la dirección `d` y que pasa por el punto `p`.
Ejemplo:
```
(sea Punto "A" libre 1 5)
(sea Recta "d1" paralela A d)
```

`prototipo (Recta nombre perpendicular p d)`

nombre : Cadena de caracteres que designa al nombre del objeto
p : Referencia a un punto
d: Referencia a una dirección (recta, segmento, vector, ...)
Devuelve: Referencia a una recta perpendicular à la dirección de `d` y que pasa por por `p`.
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Recta "d1" perpendicular A d)
```

Semirrecta

`prototipo (Semirrecta nombre 2puntos o p)`

nombre: Cadena de caracteres que designa al nombre del objeto
o: Referencia a un punto, origen de la semirrecta
p: Referencia a un punto, punto de la semirrecta
Devuelve: Referencia a una semirrecta definida por su origen y por un punto.
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Punto "O" libre 0 0)
(sea Semirrecta "dd1" 2puntos A O)
```

Segmento

`prototipo (Segmento nombre extremos p1 p2)`

nombre: Cadena de caracteres que designa al nombre del objeto
p1: Referencia a un punto
p2: Referencia a un punto
Devuelve: Referencia a un segmento definido por sus extremos
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Punto "B" libre 10 4)
(sea Segmento "" extremos A B)
```

Círculo

`prototipo (Círculo nombre 2puntos c p)`

nombre: Cadena de caracteres que designa al nombre del objeto
c: Referencia a un punto, centro del círculo
p: Referencia a un punto sobre el círculo
Devuelve: Referencia a un círculo definido por su centro y por un punto.
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Punto "B" libre 10 4)
(sea Círculo "C1" 2puntos A B)
```

`prototipo (Círculo nombre centro-radio c r)`

nombre: Cadena de caracteres que designa al nombre del objeto
c: Referencia a un punto, centro del círculo
r: Referencia a una valor numérico, radio del círculo
Devuelve: Referencia a un círculo definido por su centro y por su radio
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Número "r" libre 10)
(sea Círculo "C1" centro-radio A r)
```

`prototipo (Círculo nombre centro-segmento c s)`

nombre: Cadena de caracteres que designa al nombre del objeto
c: Referencia a un punto, centro del círculo
s: Referencia a un segmento dada la donde la longitud del segmento será el radio del círculo
Devuelve: Referencia a un círculo definido por su centro y por un segmento cuya longitud es su radio.
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Círculo "C1" centro-segmento A s)
```

Arco de círculo

`prototipo (Arco nombre 3puntos p1 p2 p3)`

nombre: Cadena de caracteres que designa al nombre del objeto
p1: Referencia a un punto, 1^{er} extremo del arco de círculo
p2: Referencia a un punto del arco
p3: Referencia a un punto, 2^{o} extremo del arco de círculo
Devuelve: Referencia a un arco de círculo definido por sus extremos y por un punto interno.
Ejemplo:

```
(sea Punto "A" libre 1 5)
(sea Punto "B" libre 0 5)
(sea Punto "C" libre -1 -2)
(sea Arco "arc" 3puntos A B C)
```

Polígono

`prototipo (Polígono nombre npuntos args)`

nombre: Cadena de caracteres que designa al nombre del objeto.
args: Una lista de referencias de puntos ; vértices del polígono.
Devuelve: Referencia a un polígono definido por sus vértices.
Ejemplo:

```
(sea Polígono "quad" npuntos A B C D)
```

Las transformaciones geométricas

Los prototipos de las transformaciones geométricas permiten la obtención de objetos rotados, reflejados, trasladados, a escala, etc.. Estos prototipos usan como referencia a los tipos punto, recta, semirrecta, vector, círculo, arco de círculo y polígono.

`prototipo (TipoDeObjeto nombre rotación objeto centro ángulo)`

TipoDeObjeto: Punto, Segmento, Recta, Semirrecta, Vector, Círculo, Arco, Polígono
nombre: Cadena de caracteres que designa al nombre del objeto
objeto: Referencia del objeto a transformar
centro: Referencia a un punto, centro de la rotación
ángulo: Referencia a un valor numérico, ángulo de la rotación
Devuelve: Referencia del objeto ya transformado.
Ejemplo:

```
(sea Punto "I1" rotación I C a)
```

`prototipo (TipoDeObjeto nombre escala objeto centro k)`

TipoDeObjeto: Punto, Segmento, Recta, Semirrecta, Vector, Círculo, Arco, Polígono

nombre: Cadena de caracteres que designa al nombre del objeto
objeto: Referencia del objeto a transformar
centro: Referencia a un punto, centro de la escala (centro de homotecia)
k: Referencia a un valor numérico, factor de la escala (homotecia)
Devuelve: Referencia del objeto transformado.
Ejemplo:
`(sea Polígono "P1" escala P C k1)`

`prototipo (TipoDeObjeto nombre simetría objeto centro)`

TipoDeObjeto: Punto, Segmento, Recta, Semirrecta, Vector, Círculo, Arco, Polígono
nombre: Cadena de caracteres que designa al nombre del objeto
objeto: Referencia del objeto a transformar
centro: Referencia a un punto, centro de la simetría (esto es, una rotación de 180 grados)
Devuelve: Referencia del objeto ya transformado.
Ejemplo:
`(sea Segmento "S1" simetría S C)`

`prototipo (TipoDeObjeto nombre reflexión objeto eje)`

TipoDeObjeto: Punto, Segmento, Recta, Semirrecta, Vector, Círculo, Arco, Polígono
nombre: Cadena de caracteres que designa al nombre del objeto
objeto: Referencia del objeto a transformar
eje: Referencia a una recta, eje de la reflexión
Devuelve: Referencia del objeto ya transformado
Ejemplo:
`(sea Polígono "P1" reflexión P d1)`

`prototipo (TipoDeObjeto nombre traslación objeto vector)`

TipoDeObjeto: Punto, Segmento, Recta, Semirrecta, Vector, Círculo, Arco, Polígono
nombre: Cadena de caracteres que designa al nombre del objeto
objeto: Referencia del objeto a transformar
vector: Referencia a un vector
Devuelve: Referencia del objeto ya transformado
Ejemplo:
`(sea Círculo "C1" traslación C v)`

Lugar geométrico

`prototipo (Lugar-geométrico nombre 2puntos m c)`

nombre: Cadena de caracteres que designa al nombre del objeto
m: Referencia a un punto móvil sobre una línea (recta, círculo, etc.)
c: Referencia a un punto fijo que depende del punto `m`
Devuelve: Referencia al lugar geométrico de `c` cuando `m` se mueve sobre la línea (recta, círculo, etc.)
Ejemplo:
`(Lugar-geométrico "locus1" 2puntos M I)`

Vector

`prototipo (Vector nombre 2puntos o e)`

nombre: Cadena de caracteres que designa al nombre del objeto
o: Referencia a un punto, origen del vector
e: Referencia a un punto, extremo del vector
Devuelve: Referencia a un vector.
Ejemplo:
```
(sea Punto "B" libre 0 5)
(sea Punto "C" libre -1 -2)
(Vector "" 2puntos C B)
```

Número

`prototipo (Número nombre libre x y v)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
v: El valor inicial del número
Devuelve: Referencia a un número libre.
Ejemplo:
```
(sea Número "pi" libre 5 5 (acos -1))
```

`prototipo (Número nombre longitud-segmento x y s)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
s: Referencia a un segmento
Devuelve: Referencia a un número, longitud de un segmento.
Ejemplo:
```
(sea Número "l" longitud-segmento 5 5 S)
```

`prototipo (Número nombre norma-vector x y v)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
s: Referencia a un vector
Devuelve: Referencia a un número, norma(longitud) de un vector.
Ejemplo:
```
(sea Número "l" norma-vector 5 5 V)
```

`prototipo (Número nombre punto-círculo x y p c)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
p: Referencia a un punto
c: Referencia a un círculo
Devuelve: Referencia a un número, distancia entre el punto y el círculo.
Ejemplo:
```
(sea Número "l" punto-círculo 5 5 P C)
```

`prototipo (Número nombre punto-recta x y p d)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
p: Referencia a un punto
c: Referencia a una recta

Devuelve: Referencia a un número, distancia entre el punto y la recta.
Ejemplo:
```
(sea Número "d" punto-recta 5 5 M D1)
```

`prototipo (Número nombre punto-punto x y p1 p2)`

nombre: Cadena de caracteres que designa al nombre del objeto
x,y: Las coordenadas del número
p1: Referencia a un punto
p2: Referencia a un punto
Devuelve: Referencia a un número, distancia entre los dos puntos.
Ejemplo:
```
(sea Número "d" punto-punto 5 5 A B)
```

`prototype (Número nombre longitud-círculo x y c)`

nombre: Cadena de caracteres que designa al nombre del objeto.
x,y: Las coordenadas del número.
c: Referencia a un círculo.
Devuelve: Referencia a un número, perímetro del círculo.
Ejemplo:
```
(sea Número "p" longitud-círculo 5 5 C)
```

`prototype (Número nombre pendiente-línea x y d)`

nombre: Cadena de caracteres que designa al nombre del objeto.
x,y: Las coordenadas del número.
d: Referencia a una recta.
Devuelve: Referencia a un número, pendiente de la recta dada.
Ejemplo:
```
(sea Número "p" pendiente-línea 5 5 d1)
```

`prototype (Número nombre longitud-arco x y arc)`

nombre: Cadena de caracteres que designa al nombre del objeto.
x,y: Las coordenadas del número.
arc: Referencia a un arco de círculo.
Devuelve: Referencia a un número, longitud del arco de círculo dado.
Ejemplo:
```
(sea Número "l"longitud-arco 5 5 ABC)
```

Ángulo

`prototipo (Ángulo nombre geométrico A B C)`

nombre: Cadena de caracteres que designa al nombre del objeto
A: Referencia a un punto
B: Referencia a un punto, vértice del ángulo
C: Referencia a un punto
Devuelve: Referencia a un ángulo geométrico
Ejemplo:
```
(sea Ángulo "a" geométrico A B C)
```

Modificación de los atributos(Apariencia) de los objetos

Para modificar los atributos de un objeto ya creado, utilizamos un sistema de envío de mensajes directamente al prototipo que representa al objeto en cuestión. La modificación de sus atributos(apariencia) es resultado de este proceso, y se hace ya que los objetos han sido creados.

`(envía objeto color valor)`

objeto: Referencia a un objeto
valor: El color; los valores posibles son: `negro, gris-oscuro, gris, blanco, verde-oscuro, verde, azul-oscuro, azul, rojo, bordeaux, amarillo, naranja`
Ejemplo:

```
(sea Punto "A" libre 1 2)
(envía A color verde)
```

`(envía línea grosor valor)`

línea: Referencia a una línea (recta, semirrecta, círculo, lugar geométrico, etc.)
valor: El grosor, los valores posibles son: `punteado, pequeño, grande`
Ejemplo:

```
(sea Punto "A" libre 1 2)
(sea Punto "O" libre 0 0)
(sea Recta "d" 2puntos A B)
(envía d grosor punteado)
```

`(envía punto tamaño valor)`

punto: Referencia a un punto
valor: El tamaño del punto, los valores posibles son: `pequeño, normal, grande`
Ejemplo:

```
(sea Punto "A" libre 1 2)
(envía A tamaño pequeño)
```

`(envía punto forma valor)`

punto: Referencia a un punto
valor: La forma del punto, los valores posibles son: `redondo , cruz, redondo-vacío,rec, rec-vacío`
Ejemplo:

```
(sea Punto "A" libre 1 2)
(envía A forma redondo)
```

`(envía objeto oculto)`

objeto: Referencia a un objeto que va a ser ocultado
Ejemplo:

```
(sea Punto "A" libre 1 2)
(envía A oculto)
```

3.3.3. Sinónimos en DGS

Esta sección consta de una transcripción ligeramente modificada del archivo `/usr/share/drgeo/scm/drgeo_scm_interface_constant_es.scm`)

Modificando este archivo se pueden obtener sinónimos para escribir en el lenguaje Scheme como si fuera Español. Esta sección ayudara a comprender programas escritos en Inglés.

Aunque es posible combinar código en Inglés con codigo con sinónimos en Español, francés o lo que se desee, convendría hacer todo el código en Inglés para asegurar su lectura por usuarios de todo el mundo.

Por ejemplo, los siguientes comandos son sinónimos:

```
(lets  Point "P" free 3 3)(send A 'color bordeaux)
(sea Punto ¨P¨ libre 3 3)(env˜a A color rojo-oscuro)
```

Notar que se han traducido varias versiones, de modo que es posible evitar escribir eñes o acentos.

```
COLORES:
negro black
gris-oscuro dark-grey
gris-obscuro dark-grey
gris grey
blanco white
verde-oscuro dark-green
verde-obscuro dark-green
verde green
azul-oscuro dark-blue
azul-obscuro dark-blue
azul blue
rojo red
rojo-oscuro bordeaux
rojo-obscuro bordeaux
amarillo yellow
anaranjado orange
naranja orange


GROSORES:
punteado dashed
pequeño small
pequeno small
grande large


PUNTOS
redondo round
redonda round
cruz cross
redondo-vacío round-empty
redondo-vacio round-empty
redonda-vacía round-empty
redonda-vacia round-empty
cuadrado-vacío rec-empty
cuadrado-vacio rec-empty
cuadrada-vacía rec-empty
cuadrada-vacia rec-empty
rec-vacía rec-empty
rec-vacia rec-empty
```

```
ESTILO
color 'color
grosor 'thickness
forma 'shape
tamaño 'size
tamano 'size
enmascarado 'masked
oculto 'masked
ocultar 'masked

PUNTOS
libre 'free
sobre-la-curva 'on-curve
sobre-curva 'on-curve
en-curva 'on-curve
mitad-2pts 'middle-2pts
mitad-segmento 'middle-segment
intersección 'intersection
intersección2 'intersection2

SEGMENTOS
extremos 'extremities
2puntos '2points

RECTAS
paralela 'parallel
ortogonal 'orthogonal
perpendicular 'orthogonal
2puntos '2points

CÍRCULOS
centro-radio 'center-radius
centro-segmento 'center-segment


ARCOS
3puntos '3points

POLÍGONOS
npuntos 'npoints

NÚMEROS
```

```
longitud-segmento 'segment-length
norma-vector 'vector-norm
longitud-vector 'vector-norm
punto-recta 'point-line
punto-círculo 'point-circle
punto-circulo 'point-circle
punto-punto 'point-point
perímetro-círculo 'circle-length
perimetro-circulo 'circle-length
longitud-círculo 'circle-length
longitud-circulo 'circle-length
pendiente-línea 'line-slope
pendiente-linea 'line-slope
longitud-arco 'arc-length

ÁNGULOS
geométrico 'geometric
geometrico 'geometric

TRANSFORMACIONES
rotación 'rotation
rotacion 'rotation
homotecia 'scale
escala 'scale
simetría 'symmetry
simetria 'symmetry
translación 'translation
translación 'translation
traslación 'translation
traslacion 'translation

GENERALES
sea lets
nueva-figura new-figure
figura-nueva new-figure
enviar send
envía send
envia send
Punto Point
Recta Line
Línea Line
Linea Line
Segmento Segment
Semirrecta Ray
Rayo Ray
Círculo Circle
Circulo Circle
Arco Arc
Arco Arc
Lugar-geométrico Locus
Lugar-geometrico Locus
Vector Vector
Numérico Numeric
Numerico Numeric
Número Numeric
Numero Numeric
Ángulo Angle
Angulo Angle
Polígono Polygon
Poligono Polygon
```

3.3.4. Galería de Ejemplos

Para ilustrar la utilización de las Figuras Scheme de Dr. Geo, proponemos una breve serie de ejemplos. En ellos mostraremos el potencial de las FSD y esperamos igualmente que le sean una fuente de inspiración. Para cada uno de estos ejemplos, damos el código fuente en Scheme de la figura, y en seguida el resultado. El código fuente puede ser copiado dentro de un archivo de texto con un editor, y luego guardado con la extensión **.scm** con lo que podrá ser evaluado después por Dr. Geo.

El código siguiente se encuentra en inglés. Favor de ver la seccion Sinónimos para poder escribir la mayor parte del código en Español.

Polígono regular

Para construir un polígono regular, con un número arbitrario de lados, podemos hacer uso de una función recursiva en Scheme. Construire un polygone régulier, avec un nombre de coté arbitraire, peut se faire par l'intermédiaire d'une fonction récursive Scheme.

```
(define pi (acos -1))
(define n 15)
(define x0 0)
(define y0 0)
(define p1 0)
```

```
(define (polygon center p a n)
  (if (> n 0)
      (begin
        (set! p1 (Point "" rotation p center a))
        (send p1 masked)
        (Segment "" extremities p p1)
        (polygon center p1 a (- n 1)))))

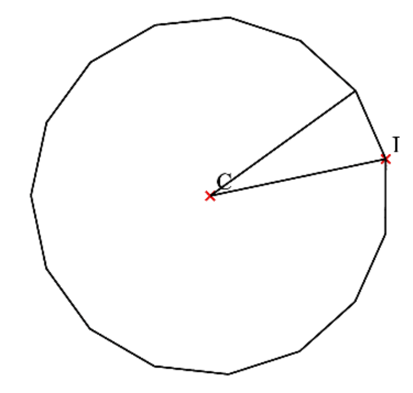
(new-figure "Regular Polygon!")
(lets Point "C" free x0 y0)
(lets Numeric "a" free 0 0 (* 2 (/ pi n)))
(send a masked)

(set! p1 (Point "I" free 5 0))

(lets Segment "S" extremities C p1)
(Segment ""  rotation S C a)

(polygon C p1 a n)
```

Figura 3.12: Un polígono regular de 15 lados



Fractal

La construcción de una curva fractal con la forma de un árbol se logra fácilmente con una Figura Scheme. El código fuente de la figura es sorprendentemente breve, sobre todo cuando lo comparamos con la construcción "a mano" usando la interfaz gráfica.

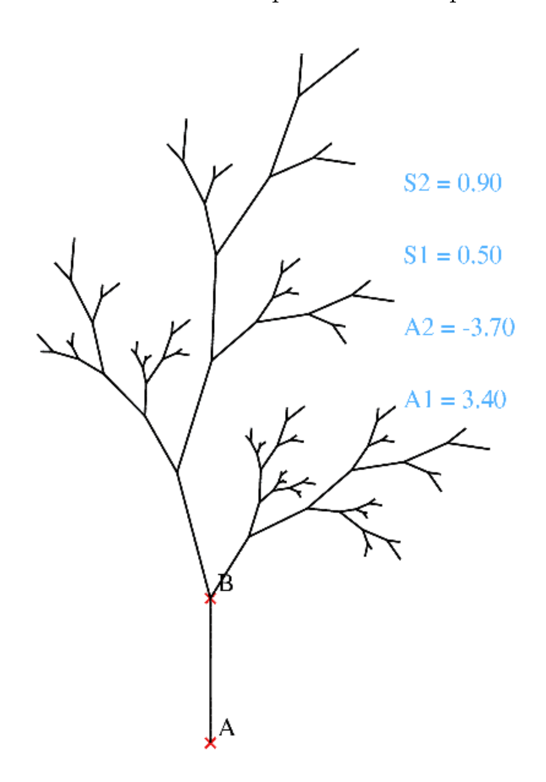
```
(new-figure "Baum")
(lets Numeric "A1" free 2 2 +3.4)
(lets Numeric "A2" free 2 3 -3.7)
(lets Numeric "S1" free 2 4 +0.5)
(lets Numeric "S2" free 2 5 +0.9)

(define (dec n)
  (- n 1))
(define (inc n)
  (+ n 1))
```

```
(define (invisible p)
  (send p masked)
  p)
(define (scalerot oP C a s)
  (let* ((sP (invisible (Point "" scale oP C s)))
         (rP (invisible (Point "" rotation sP C a))) )
    rP))
(define (Zweig p0 p1 n)
  (Segment "" extremities p0 p1)
  (let* ((left-scale (if (odd? n) S1 S2))
         (left-angle A1)
         (right-scale (if (odd? n) S2 S1))
         (right-angle A2) )
    (if (> n 0)
      (begin
        (Zweig p1 (scalerot p0 p1  left-angle  left-scale) (dec n))
        (Zweig p1 (scalerot p0 p1 right-angle right-scale) (dec n))))))

(lets Point "A" free -3 0)
(lets Point "B" free -3 2)
(Zweig A B 6)
```

Figura 3.13: Una curva fractal que simula la representación de un árbol



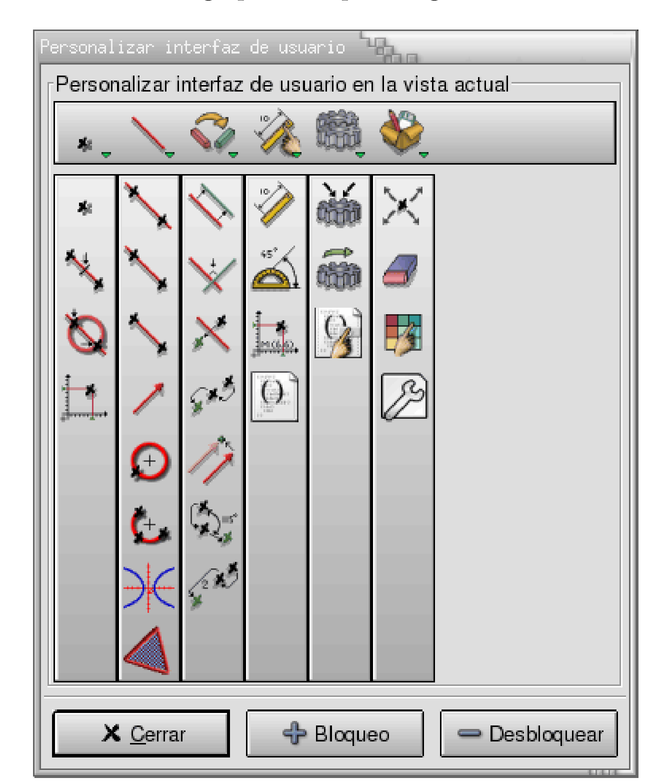
3.4. Bloquear las herramientas de la interfaz

Dr. Geo ofrece la posibilidad de preparar sesiones[10] dentro de los cuales el maestro puede decidir, para ciertas figuras, impedir el acceso a ciertas herramientas, el bloqueo se realiza utilizando una contraseña distinta para cada figura. Esto da al maestro flexibilidad al preparar una actividad con distintas figuras, y diversas herramientas disponibles en cada figura.

3.4.1. Bloqueo de las herramientas

Para impedir el acceso a las herramientas de construcción recurrimos al comando `Editar->Interfaz personalizado`. Después de haberlo seleccionado se abre una gran ventana de diálogo con todos los iconos de los instrumentos de Dr. Geo. Allí, podemos hacer click sobre los iconos para activarlos o desactivarlos. Cuando una herramienta está desactivada, el icono que la representa aparece ligeramente desvanecido. Es igualmente posible desactivar todas las herramientas en una misma columna haciendo click en el icono principal del menú (que se reconoce por tener un pequeño triángulo verde en la parte inferior derecha).

Figura 3.14: Ventana de diálogo para bloquear algunas herramientas de la interfaz



Cuando el usuario ha terminado de seleccionar los iconos que han de ser bloqueados, se procede al bloqueo propiamente dicho haciendo click sobre el botón `bloqueo` en la ventana de diálogo. En este momento, Dr. Geo pedirá una contraseña.

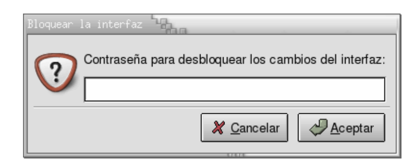
(!) Cuando se guarda una figura o una sesión en que se impide el acceso a ciertas herramientas, las contraseñas son igualmente guardadas, de forma encriptada, en el archivo

[10] Archivos que contienen varios documentos de Dr. Geo (i.e. figuras y/o texto)

3.4.2. Desbloqueo de las herramientas

Es por supuesto posible desbloquear la interfaz, ya sea para dar acceso progresivo a las herramientas – estando en la clase con los estudiantes– o para volver a diseñar la figura. Para ello, se necesita escoger dentro de la ventana de diálogo el botón `Desbloquear`. Dr. Geo pide entonces la contraseña que se ingresó previamente.

Figura 3.15: La ventana de diálogo para desbloquear la interfaz de una figura.



Capítulo 4

Archivos y documentos

Las construcciones pueden ser guardadas de dos maneras. Una construcción por archivo o un conjunto de construcciones por archivo (i.e. una sesión de Dr. Geo). Le recordamos que los documentos que con tienen las figuras debes ser guardados con una extensión **.fgeo** y que aquellos que contengan sólo macros que sean guardados con la extensión **.mgeo**. Para los documentos que contienen varias figuras, construcciones de macros y textos explicativos, utilizar la extensión **.fgeo**

Lo anterior es sólo indicativo, pero seguir estas reglas permitirá identificar y volver a cargar los archivos más fácilmente.

4.1. Guardar una construcción

A partir del menú `Archivo->Guardar` o del menú `Archivo->Guardar como...`, un archivo que contiene la figura de la vista activa puede ser guardada.

(!) Dr. Geo puede trabajar con varias figuras al mismo tiempo. El usuario puede pasar de una figura a otra haciendo click sobre el nombre de la figura (ubicado en la parte baja de la figura).

.

Con el segundo menú, el usuario puede cambiar el nombre del documento a guardar.

(!) El nombre del archivo inicialmente propuesto puede ser cambiado con el menú`Editar->Preferencias...`. Para más información ver la sección comportamiento predeterminado Sección 2.3.1, página 21.

4.2. Guardar una sesión

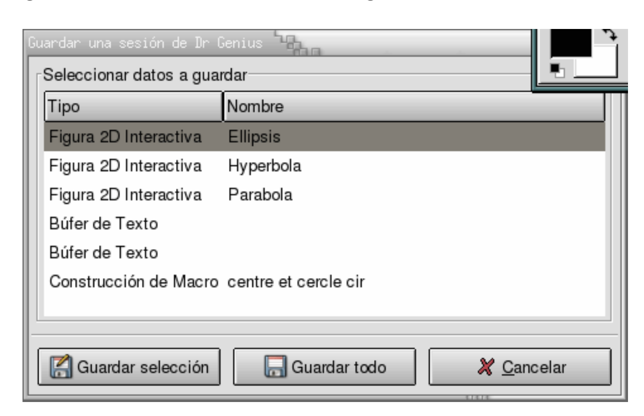
Una sesión es un conjunto de objetos de Dr. Geo que el usuario desea guardar de un solo golpe en un archivo. Esto permite al maestro organizar un conjunto de objetos (figuras, macros y notas de clase) en un solo archivo con el fin de facilitar su uso.

A partir del menú `->Guardar múltiple`, el usuario puede abrir la ventana de diálogo de la sesión.

En esta ventana de diálogo, la lista de todos los objetos activos está presente en una tabla. La primera columna muestra los tipos de objetos contenidos en la sesión de Dr. Geo, la segunda muestra los nombres de los objetos.

(!) De hecho, una sesión puede contener tres tipos de objetos : figuras interactivas en dos dimensiones, macros y textos.

Figura 4.1: La ventana de diálogo de la sesión Dr. Geo



El usuario puede seleccionar uno por uno los objetos a guardar y después hacer click en el botón`Guardar selección`. Alternativamente, puede guardar todos los objetos haciendo click sobre el botón `Guardar todo`.

(!) El menú `Archivo->Guardar múltiple` es el único medio de guardar una macro en un archivo.

4.3. Guardar una Macro

Para guardar una o varias macros en un archivo. es necesario proceder como cuando uno desea guardar una sesión –guardar múltiple–. Posteriormente , en la ventana para guardar una sesión, seleccionar la o las macros que se desea guardar. En seguida guárdelas en un archivo que contenga la extensión **.mgeo**. ¡Eso es todo!

Es de este modo posible crear bibliotecas de macros, una por archivo, o varias agrupadas por temas en un sólo archivo.

4.4. Abrir un archivo

Ya sea que el usuario haya guardado una sola figura o una sesión con los distintos objetos, el procedimiento para la apertura de cualquier tipo de archivo es siempre a través del menú `Archivo->Abrir`. Si la sesión recién abierta contiene macros, estas están directamente disponibles a partir de que la herramienta para ejecutar macros esté disponible para la ejecución. Las macros están disponibles para todas las figuras abiertas.

4.5. Exportar una figura

Dr. Geo ofrece la posibilidad de exportar una figura geométrica a un documento LaTeXo PostScript. Estos dos formatos de exportación son de tipo vectorial, a diferencia de las imágenes de mapas de bits que son de menor calidad para una impresión. Los comandos de exportación son accesibles a través del menú `Archivo->Exportar como...`

4.5.1. Exportar a LaTeX

En el caso de una exportación al formato LaTeX, el documento exportado necesita el paquete `pstricks`. Éste es en general distribuido con LaTeX. Típicamente un documento exportado en LaTeX podrá ser integrado dentro de otro documento LaTeX o bien directamente compilado con los comandos:

```
latex figure.tex
dvips figure.dvi
```

Esta serie de comandos permite obtener el documento figure.ps, el cual puede ser abierto por el programa GhostView[1] `gv`.

4.5.2. Exportar a PostScript

La exportación al formato PostScript – extensión `eps` – ofrece la ventaja de ser fácilmente utilizable por diversos programas [2]. En efecto, el formato EPS es cercano a ser un estandard en lo que se refiere a imágenes vectoriales. Para visualizar rápidamente este tipo de imágenes,utilizamos, como ya antes mencionamos, el programa GhostView; en una terminal se invoca escribiendo `gv`.

4.5.3. Exportar a png

Ciertos programas como OpenOffice.org sólo explotan parcialmente el formato EPS[3].

Es posible, afortunadamente, exportar al formato PNG. La imagen exportada bajo este formato tiene una alta definición y posee un fondo transparente. Esto puede permitir escribir un texto a lo largo de la figura.

4.5.4. Exportar Fly Draw

El servidor de ejercicios WIMS desarrollado por el Dr. Xiao Gang en la universidad de Nice dispone de un formato de descripción de figuras llamado Fly Draw. DR. GEO puede exportar hacia ese formato.

4.5.5. Definir el área de exportación

Inicialmente, DR. GEO exporta la zona visible de la figura. Por lo anterior, si deseamos exportar una zona precisa de la figura, una manera de hacerlo es redimensionar la ventana de DR. GEO hasta obtener la zona deseada [4].

Esta solución, empero, aunque rápida y simple, no es lo suficientemente flexible. Además, el área de exportación no queda guardada junto con la figura. Afortunadamente, existe otro modo de definirla usando el comando `Definir el área de exportación` el cual es accesible desde el menú `Archivo->Preferencias de Exportación`

Habiendo seleccionado el comando, podemos seleccionar un rectángulo en la figura que corresponderá con el área de exportación. Esta área puede ser redefinida tantas veces como sea necesario. En la figura estará representada por un rectángulo gris claro. Cuando la figura sea guardada, esta zona lo será también.

Para suprimir el área de exportación y regresar al comportamiento inicial de DR. GEO use el comando `Eliminar área de exportación`, el cual, por supuesto, no borra ningún objeto de la figura.

---

[1] GhostView es un programa que permita visualizar documentos PostScript (.ps o .eps) así como documentosPDF

[2] Este formato es reconocido por TeXmacs, OpenOffice.org, Lyx, LaTeX Xfig, el Gimp y muchos más.

[3] OpenOffice.org permite integrar una imagen EPS pero no la puede mostrar. Únicamente en la impresión en papel aparecerá la imagen

[4] Cuando lo deseemos, el panel lateral que contiene el árbol lógico de la figura, puede ser desplegado reduciendo aún más la zona visible, y, en consecuencia, el área de exportación.

Figura 4.2: Una figura con un área de exportación definida

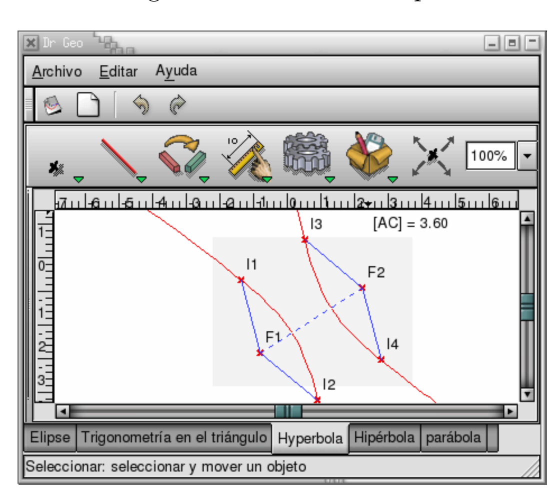
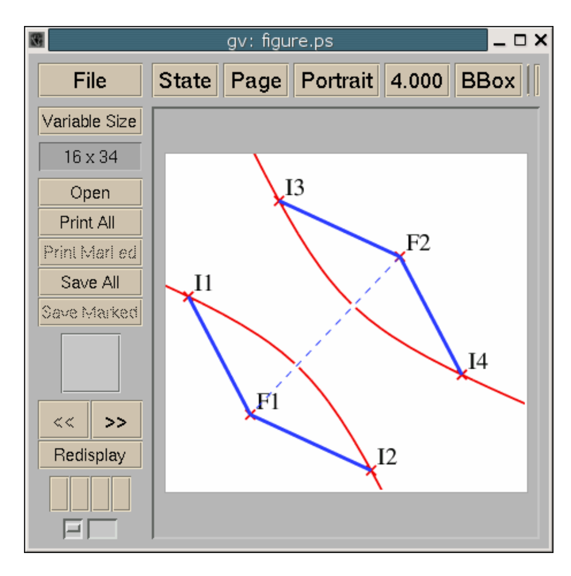


Figura 4.3: El área de exportación ha sido exportada a un documento PostScript; es visible con el programa GhostView



Capítulo 5

Aplicaciones Didácticas

Este capítulo constituye una ayuda para el usuario que desea familiarisarse con Dr. Geo a través de ejemplos. A diferencia de los capítulos precedentes, el enfoque es más concreto, y es orientado a situaciones precisas. El capítulo consiste en diversas aplicaciones pedagógicas.

5.1. Pitágoras y scripts

Una de las aplicaciones didácticas en Dr. Geo consiste en la utilización de los scripts Scheme Seccióon 3.2, página 29 para resolver ejercicios de Geometría.

Como ejemplo, vamos a mostrar la solución a un problema clásico en cuya solución se usa el teorema de Pitágoras. El texto es el siguiente :

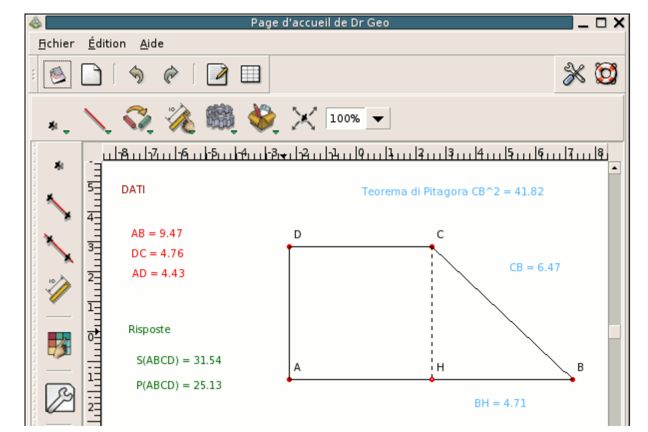
> *Sea un trapecio rectángulo ABCD del que son conocidas las bases y la altura. Calcular el perímetro y el área del trapecio.*

No es difícil, siguiendo el mismo modelo, desarrollar otros ejemplos semejantes.

Solución :

Comencemos por construir la figura en Dr. Geo, la cual debe ser como sigue :

Figura 5.1: Trapecio Rectángulo



La figura contiene los datos a partir de los cuales podemos resolver el problema. Podemos contestar enseguida la pregunta relacionada con el área, para ello podemos escribir el script Scheme sigueinte que tiene como entradas las dos bases y la altura del trapecio :

```
define AB (getLength a1))
(define DC (getLength a2))
(define AD (getLength a3))
(/ ( * AD (+ AB DC )) 2 )
```

Calculamos la longitud del segmento BH escribiendo un script Scheme que tiene por parámetros de entrada a los segmentos AB y CD. El texto del script es el siguiente :

```
(define AB (getLength a1))
(define CD (getLength a2))
(- AB CD)
```

En este momento, podemos aplicar el Teorema de Pitágoras al triángulo rectángulo CHB. Aquí también utilizamos un script Scheme que tiene por parámetros de entrada al segmento CH y al script BH :

```
(define CH (getLength a1))
(define BH (getValue a2))
(+ (* CH CH) (* BH BH))
```

Finalmente, podemos obtener el valor del segmento BC calculando la raiz cuadrada del valor devuelto por el script anterior :

```
(define q (getValue a1))
( sqrt q )
```

Los dos scripts anteriores pueden ser combinados en un solo script un poco más elaborado.
Ahora podemos concluir el ejercicio calculando el perímetro con un script Scheme :

```
(define AB (getLength a1))
(define CB (getValue  a2))
(define DC (getLength a3))
(define AD (getLength a4))
(+ (+ AB CB )(+ DC AD ))
```

5.2. Teoremas y conjeturas

Los scripts Scheme Secciónn 3.2, página 29 permiten no sólo resolver ejercicios, sino también comprender mejor el enunciado de los teoremas (y las hipótesis de los mismos) asi como corroborar o descartar conjeturas.
En esta sección comenzaremos analizando el enunciado del Teorema de Ptolomeo :

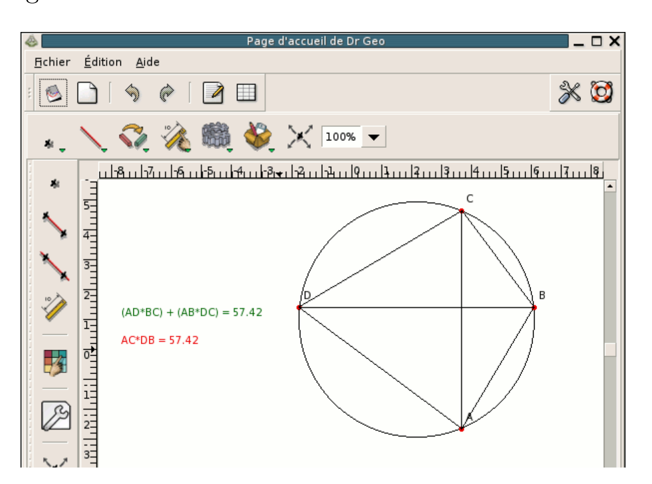
Dado un cuadrilátero inscrito en una circunferencia, la suma de los productos de los lados opuestos es igual al producto de las diagonales.

Podemos construir la figura con Dr. Geo como abajo :
En la figura, hemos implementado dos scripts que calculan, respectivamente, la suma del producto de los lados opuestos, y el producto de las diagonales.
El primer script es el siguiente :

```
(define AB (getLength a1))
(define DC (getLength a2))
(define BC (getLength a3))
(define AD (getLength a4))
(+ (* AB DC )(* BC AD ))
```

El segundo script es .

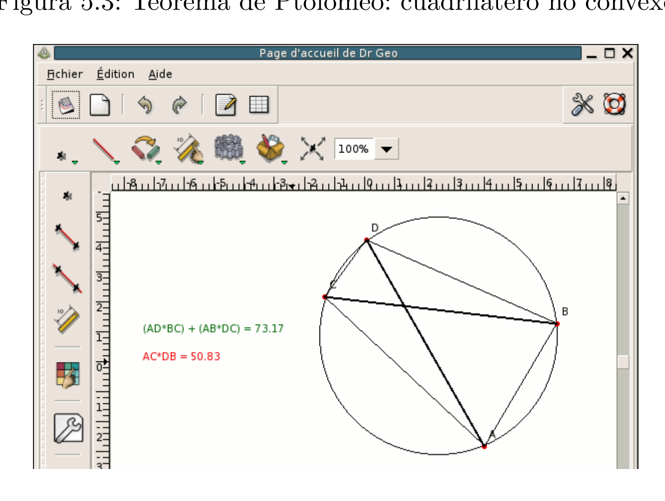
Figura 5.2: Teorema de Ptolomeo: cuadrilátero convexo




```
(define DB (getLength a1))
(define AC (getLength a2))
(* DB AC )
```

Como podemos ver, los valores devueltos por los dos scripts, en conformidad con el Teorema de Ptolomeo, son iguales [1]. Cuando modificamos dinámicamente la figura, los valores de los scripts son idénticos, salvo en el caso siguiente :

Figura 5.3: Teorema de Ptolomeo: cuadrilátero no convexo



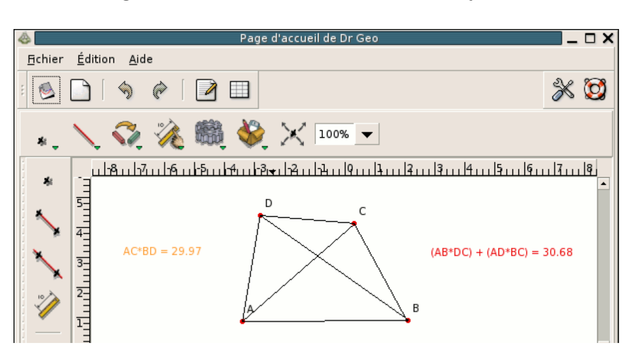
Vemos en la figura que el cuadrilátero ya no es convexo.

En este caso el enunciado del Teorema es falso. Por lo tanto, para contemplar ese caso, debemos reformular el enunciado como sigue :

Dado un cuadrilátero CONVEXO inscrito en una circunferencia, la suma del producto de lados opuestos es igual al producto de las diagonales..

En este momento las conjeturas aparecen de modo natural : ¿es acaso valida la conclusión del teorema de Ptolomeo también valida para un cuadrilátero convexo no inscrito en un círculo ?(i.e. ¿es cierto el recíproco?)

Figura 5.4: Refutación de la conjetura



Con Dr. Geo podemos ver que dicha conjetura es falsa como muestra la figura siguiente :

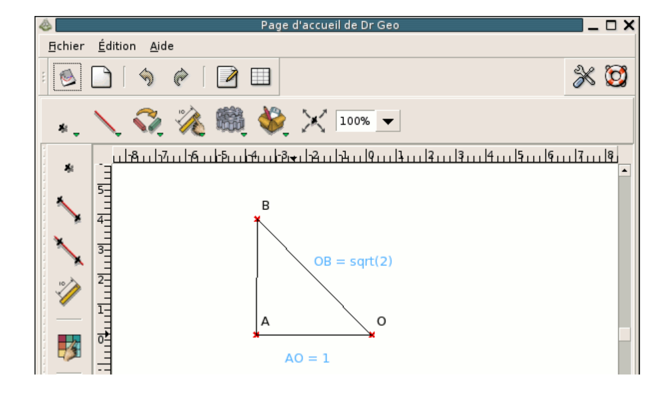
El lector no tendrá problemas en utilizar Dr. Geo para la construcción de ejemplos didácticos, relacionados probablemente con teoremas más conocidos, como el Teorema de Pitágoras o alguno de los teoremas de Euclides.

5.3. Números irracionales

Una construcción clásica, relacionada con los números irracionales, conocida bajo el nombre de "Espiral de Teodoro" permite construir geométricamente la raíz cuadrada de números enteros a partir de un triángulo isósceles.

Consideremos el triángulo OAB donde $OA = 1$:

Figura 5.5: Construción de la raíz cuadrada de 2



Por el teorema de Pitágoras, tenemos que OB es igual a la raíz cuadrada de 2. Si, ahora, con la figura, construímos un nuevo triángulo rectángulo en B, con lados OB y BC tal que $BC = 1$.

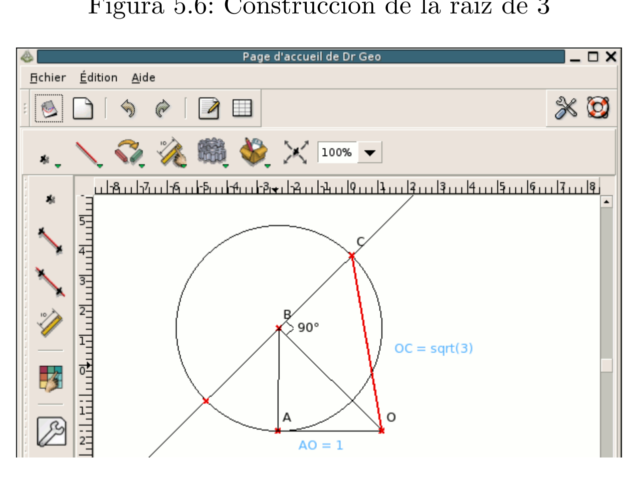
Nuevamente, por el teorema de Pitágoras, es claro que la hipotenusa OC del triángulo OBC tiene por longitud la raíz cuadrada de 3. Repitiendo este proceso obtenemos todas las raíces cuadradas de los números naturales.

La naturaleza repetitiva de la construcción se presta perfectamente para la utilización de un FSD. Consideremos el código siguiente :

```
(new-figure "Triangle")
```

[1] Esto es una verificación numérica; no una prueba

Figura 5.6: Construcción de la raíz de 3




```
(define (triangle p1 p2 p3 n)
  (let* ((s1 (Segment "" extremities    p1 p2))
         (s2 (Segment "" extremities    p2 p3))
         (s3 (Segment "" extremities    p3 p1))
         (pe (Line    "" orthogonal     p3 s3))
         (ci (Circle  "" center-segment p3 s2))
         (p4 (Point   "" intersection2  pe ci)))
    (send pe masked)
    (send ci masked)
    (send p4 masked)
    (if (> n 0)
       (triangle p1 p3 p4 (- n 1)))))


(lets Point "O" free  0  0)
(lets Point "A" free -1  0)
(lets Point "B" free -1  1)
(triangle O A B 15)
```

El triángulo inicial está definido a través de coordenadas por comodidad unicamente. El código es la transcripción literal del procedimiento repetitivo previamente descrito. Una vez evaluado por Dr. Geo, el código nos da la siguiente figura :

Las hipotenusas de cada triángulo tienen por longitud las raíces cuadradas de los números enteros naturales entre 2 y 17.

5.4. Espiral de Baravelle

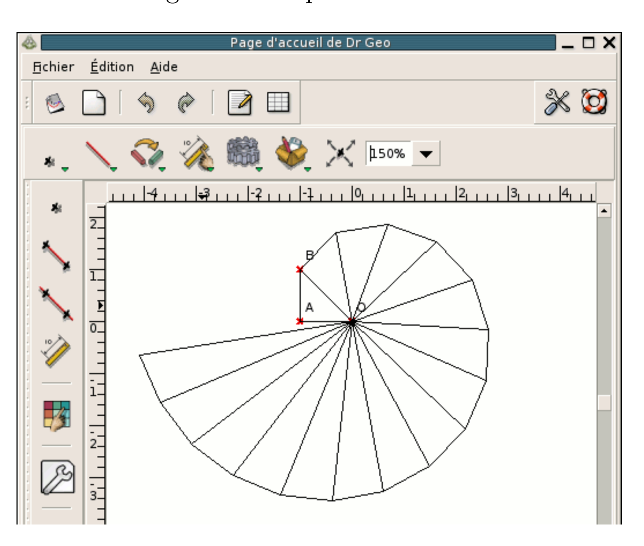
Como vimos anteriormente, con la ayuda de una FSD es posible construir de manera intuitiva y simple figuras que permiten visualizar situaciones que en un programa son ya sea iterativas o recursivas.

Profundicemos un poco más en este aspecto. Modificando el código Scheme utilizado para la construcción de los números irracionales, podemos obtener una figura famosa en la literatura matemática: la espiral de Baravelle.

El código Scheme que define la espiral es el siguiente :

```
(new-figure "Baravelle")
```

Figura 5.7: Espiral de Teodoro




```
(define (triangle p1 p2 p3 n)
  (let* ((s1  (Segment "" extremities    p1 p2))
         (s2  (Segment "" extremities    p2 p3))
         (s3  (Segment "" extremities    p3 p1))
         (m   (Point   "" middle-2pts    p1 p3))
         (r   (Segment "" extremities    m  p3))
         (pe  (Line    "" orthogonal     p3 s3))
         (ci  (Circle  "" center-segment p3 r ))
         (p4  (Point   "" intersection2  pe ci)))
    (send pe  masked)
    (send ci  masked)
    (send p4  masked)
    (send m   masked)
    (if (> n 0)
      (triangle  m  p3  p4 (- n 1)))))


(lets Point "A" free  0  5)
(lets Point "B" free  5  5)
(lets Point "C" free  5  0)
(triangle A B C 9)

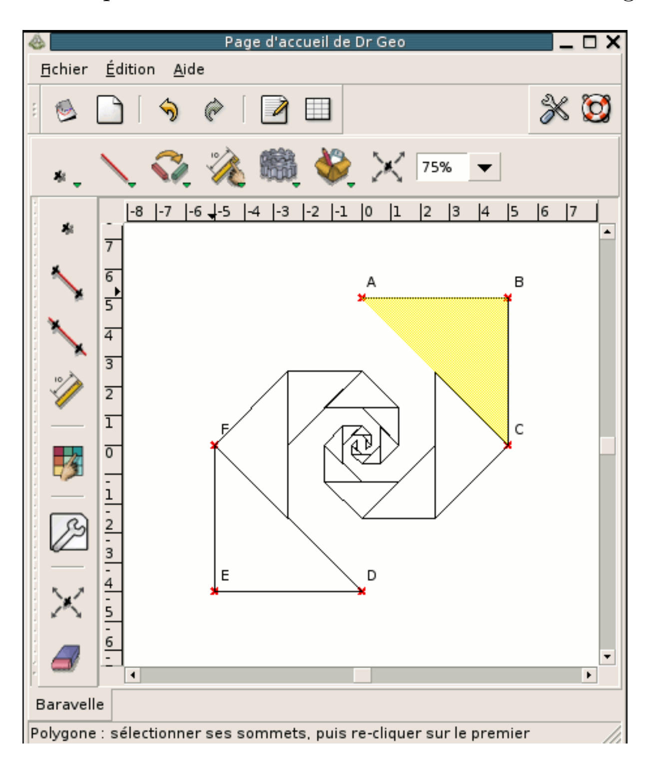
(lets Point "D" free  0 -5)
(lets Point "E" free -5 -5)
(lets Point "F" free -5  0)
(triangle D E F 9)
```

Con base en la figura y en el código Scheme correspondiente, vemos claramente la naturaleza iterativa del mecanismo de construcción de la figura. Un problema interesante, que dejamos al lector, consiste en establecer cuándo las dos ramas de la espiral convergen.

Una pequeña variación al codigo precedente :

```
(new-figure "Spirale")
```

Figura 5.8: La espiral de Baravelle obtenida al evaluar el código Scheme



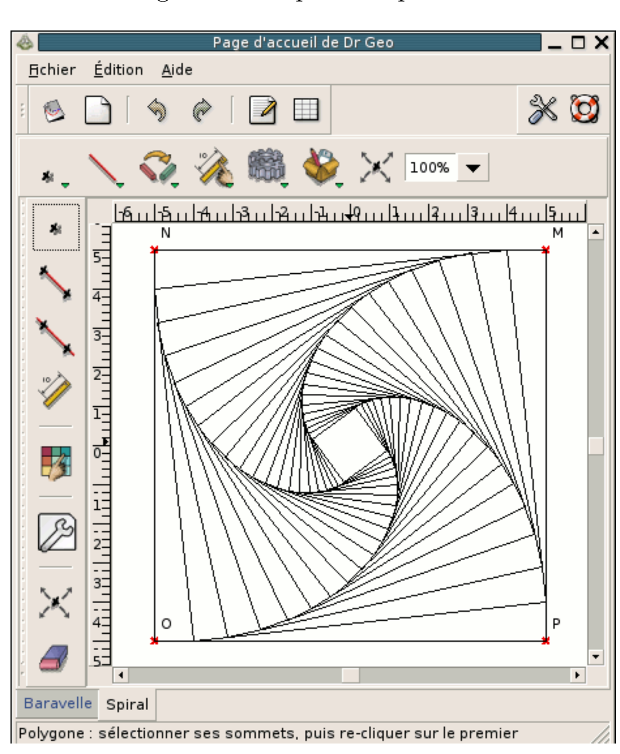

```
(define (square p1 p2 p3 p4 n)
  (let* ((s1 (Segment "" extremities p1 p2))
         (s2 (Segment "" extremities p2 p3))
         (s3 (Segment "" extremities p3 p4))
         (s4 (Segment "" extremities p4 p1))
         (A (Point "" on-curve s1 1/10))
         (B (Point "" on-curve s2 1/10))
         (C (Point "" on-curve s3 1/10))
         (D (Point "" on-curve s4 1/10)))
     (send A masked)
     (send B masked)
     (send C masked)
     (send D masked)
     (if (> n 0)
       (square A B C D (- n 1)))))

(lets Point "M" free 5 5)
(lets Point "N" free -5 5)
(lets Point "O" free -5 -5)
(lets Point "P" free 5 -5)

(square M N O P 30)
```

Obtenemos, pues, una espiral simplificada.
El lector está invitado a divertirse creando nuevas variaciones.

Figura 5.9: Espiral simplificada

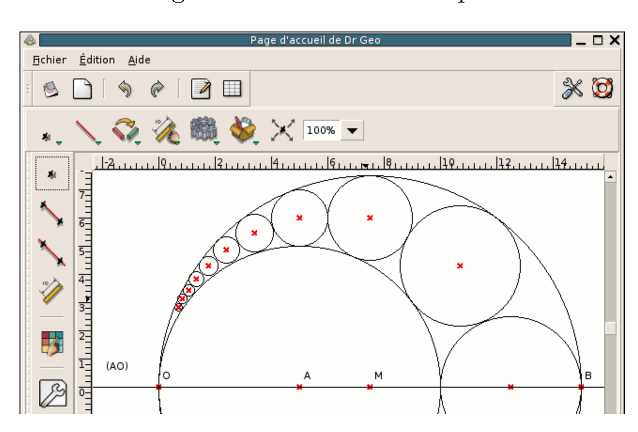


5.5. Cadena de Papus

Una aplicación de las Figuras Scheme de Dr. Geo consiste en la reproducción de una figura cuando sólo conocemos sus características analíticas.

Nos proponemos desarrollar un ejemplo famoso: “La Cadena de Papus”.

Figura 5.10: Cadena de Papus



Los centros y los radios de los círculos que la constituyen tienen una expresión analitica conocida, y es sencillo implementar la expresión analítica en una FSD que reproducirá la figura.

```
(new-figure "Pappo")

(define (circle n)
(let*(
  (r (Numeric "" free 0 0 (/ 15 ( + 6 (* n n )))))
  (c (Point   "" free (* 5 (/ 15 ( + 6 (* n n ))))
                      (* 2 (* n (/ 15 ( + 6 (* n n )))))))
  (p (Circle "" center-radius c r )))
 (send r masked)
 (if (> n 0)
    (circle (- n 1)))))

(circle 8)

(lets Point "A" free 5 0)
(lets Point "O" free 0   0)
(lets Point "B" free 15 0)
(lets Point "M" middle-2pts B O)
(lets Circle "" 2points M O)
(lets Circle "" 2points A O)
(lets Line   "" 2points A O)
```

El código de la figura es relativamente intuitivo y no necesita mayor comentario.

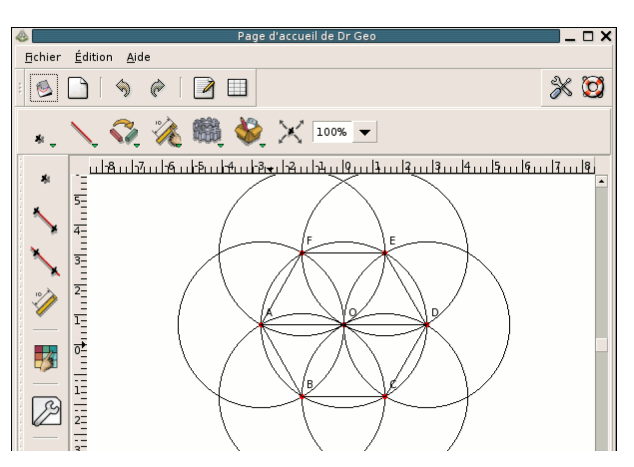
Un ejercicio no trivial, que dejamos al lector, consiste en determinar una construcción con regla y compás que conduzca a una implementación iterativa. (la implementación dada es recursiva).

5.6. Cálculo de π

El cálculo aproximado de π ha tenido un papel importante en la historia de las Matemáticas. Los métodos para su cálculo son diversos, teniendo mejoras de uno a otro método. Nos proponemos atacar el problema con un método que llamaremos, aunque no sea del todo apropiado, **Método de Exhaución**. Éste método tiene la ventaja de mostrar la esencia misma del problema. (El método siguiente fue desarrollado por Arquímides usando el método de Exhaución desarrollado por Eudoxo, el cual fue precursor de la teoría de límites).

Comenzaremos con la construcción de un hexágono regular inscrito en una circunferencia a partir de su lado BC. Notemos de paso que es posible a partir de esta construcción crear y memorizar una macro que llamaremos *Hexágono*

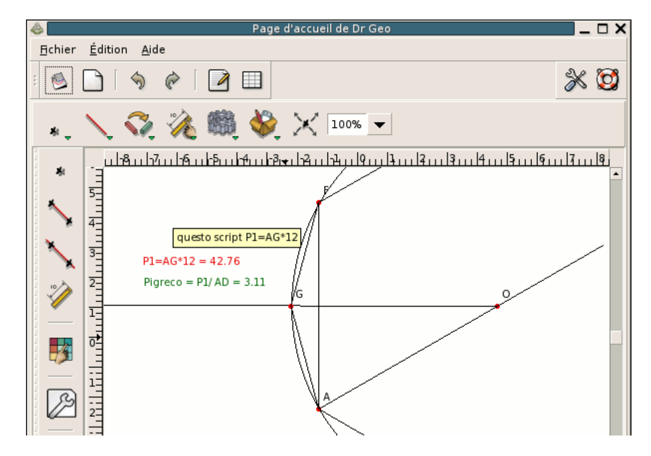
Figura 5.11: Hexágono regular inscrito



La idea del método de exhaución consiste en primero aproximar la longitud del círculo con el perímetro $P(0)$ del hexágono y de calcular una aproximación de π dividiendo $P(0)$ por el diámetro del círculo. Claramente, la aproximación obtenida sera de 3.

En una segunda etapa podemos, utilizando Dr. Geo, construir dentro de la misma circunferencia, un dodecágono regular. Calculemos su perímetro,$P(1)$, y dividamos por el diámetro de la circunferencia. Esto nos dará una mejor aproximación.

Figura 5.12: Aproximación de π



Duplicando en cada paso el número de lados del polígono regular inscrito obtenemos

mejores aproximaciones.

Capítulo 6

Libro de recetas

Este capítulo constituye una ayuda para el usuario que desea realizar construcciones geométricas con Dr. Geo. Contrariamente a los capítulos precedentes, el enfoque es hacia problemas precisos. El contenido de este capítulo es fruto del uso que los usuarios han hecho del programa, y responde a algunas peticiones de los mismos.

6.1. Crear un polígono regular

Dr. Geo puede crear varias macros con elementos independientes (i.e. varios puntos libres, valores libres o scripts independientes). Para hacer esto drgeo dota de valores a los parámetros libres de esos objetos [1]. Dr. Geo puede crear varias macros. El o los valores escogidos son, en efecto, los valores del objeto que sirvió de modelo durante la creación de la macro.

En lo que sigue, un pequeño ejemplo que permite crear polígonos regulares de cualquier número de lados es presentado. En varias de las macros, un script Scheme interviene en el proceso.

El ejemplo es mostrado paso a paso, aunque el lector no familiarizado con las macros, los scripts o más generalmente el funcionamiento de Dr. Geo está invitado a leer las secciones anteriores.

6.1.1. El modelo del polígono

Para comenzar, crearemos un punto libro, en seguida un Script que no toma parámetros de entrada (para crear el script, seleccionar la herramienta script y hacer enseguida click sobre el fondo de la figura). Editar el script y escribir lo siguiente :

```
(/ (* 2 (acos -1)) 7)
```

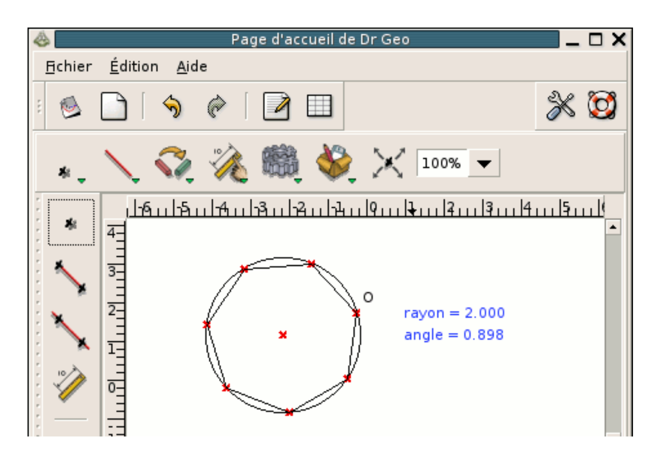
El resultado de este script es el valor del script y servirá para construir, a través de una rotación un polígono regular de 7 lados. Colocar enseguida un valor libre con valor 2. Éste será el radio de nuestro polígono.

Construir un círculo de centro el punto y de radio el valor recién creado. Sobre este círculo colocar un punto. Llamemos a este punto O.

Ahora utilicemos la herramienta rotación para crear la imágen del punto O bajo la rotación con centro de rotación en el centro del círculo, y con un ángulo de rotación cuyo valor es el valor del script. Continuar sucesivamente hasta obtener los siete vértices del polígono. Unir los vertices con los segmentos hasta obtener el polígono.

[1]Si no, las macros no podrían ser llamadas. Por ejemplo,¿qué valor debemos dar a un objeto tipo valor libre que interviene dentro de la construcción de una figura?

Figura 6.1: Resultado de construir un polígono regular de 7 lados.



6.1.2. La macro del polígono

Abrir el diálogo para la creación de la macro y crearla como se indica a continuación :

1. **parámetro de entrada :** el centro del círculoe ;
2. **paramètres de sortie :** el punto O y los 7 lados del polígono.

Listo, la macro ha sido creada.

Para crear un polígono regulara de 7 lados, abrir la ventana de diálogo para ejecutar una macro. Basta seleccionar un punto de la figura para crear un polígono de 7 lados.

6.1.3. Algunas consideraciones

Cuando Dr. Geo encuentra los elementos libres en una macro, fija sus valores toamndo a la figura como modelo. En nuestro ejemplo, el script, el valor del radio y el punto O son los elementos libres de los que hablamos. Cuando la macro es ejecutada, sus valores se fijan como sigue :

- El valor se fija 2 ;
- el script se fija en `(/ (* 2 (acos -1)) 7)` ;
- el punto sobre el círculo se fija con la misma abscisa curvilínea que la que tenía el punto O..

Es esto último lo que da sentido a la utilización conjunta de macros con scripts u otros elementos libres.

En otras palabras, por ejemplo, si para construir una figura necesitamos 7 elementos libres al empezar, y al crear la macro sólo damos 1, el resto de los elementos libres necesarios para la figura se toman de la figura que sirvió de modelo.

6.2. Imprimir una figura

Lo más sencillo es exportar la figura al formato PostScript – Exportar a PostScript Sección 4.5.2, página 55. En seguida es posible usar un programa que permita visualizar los documentos e imprimirlos. El documento, de calidad vectorial, puede ser abierto e impreso por programas como GhostView, Gnome GhostView o incluso K GhostView, por citar algunos.

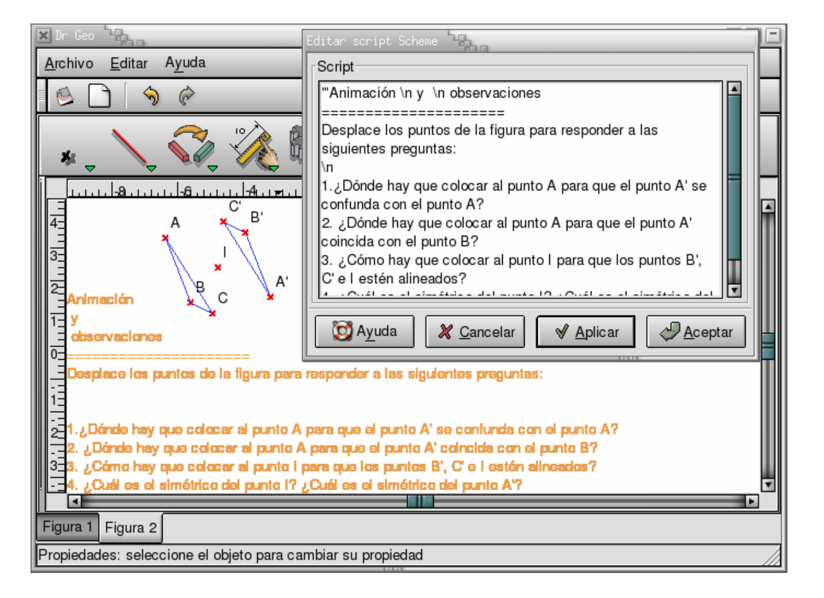
6.3. Colocar un párrafo de texto en la figura

Uno puede ventajosamente utilizar los scripts, de una manera ciertamente imprevista. Se selecciona la creación de un script –Herramienta Script Sección 2.1.4, página 16 –, y se hace click sobre el fondo de la figura. De este modo un script sin parámetros de entrada es creado.

Enseguida se editan las propiedades del script – Propiedades de un objeto Sección 2.2.5, página 20. En la zona de texto donde el código Scheme es normalmente escrito, colocamos un texto entre comillas precedido por una única apóstrofe, por ejemplo :

```
'"Animación \n  y \n  observaciones
=====================
Desplace los puntos de la figura para responder a las siguientes preguntas:
\n
1.>Dónde hay que colocar al punto A para que el punto A' se confunda con el punto A?
2. >Dónde hay que colocar al punto A para que el punto A' coincida con el punto B?
3. >Cómo hay que colocar al punto I para que los puntos B', C' e I estén alineados?
4. >Cuál es el simétrico del punto I? >Cuál es el simétrico del punto A'?"
```

Figura 6.2: Ejemplo de párrafo de texto dentro de una figura



Como con todo script, es igualmente posible cambiar el color del texto. Observe que es posible usar algunos caracteres de control dentro del texto. Un script, que permite combinar cálculos con texto es el siguiente :

```
(format #f "~s   más   ~s es casi siempre ~s" 3 4 (+ 3 4))
```

Índice de figuras

1.1. Pantalla de Bienvenida de Dr. Geo 8
1.2. Una figura geométrica vacía 8
1.3. Una figura de Dr. Geo y su descripción 9

2.1. Categorías de herramientas de Dr. Geo y sus descripciones 11
2.2. El diálogo de estilo para un objeto tipo punto 18
2.3. El diálogo de apariencia para objetos tipo línea 19
2.4. El diálogo de estilo para objetos numéricos y tipo polígono. 19
2.5. Editar las coordenadas de un punto libre 20
2.6. Editar valor libre 20
2.7. Editando un script 20
2.8. Preferencias de las figuras geométricas 21
2.9. Renombrar la vista de una figura 22

3.1. Nuestra figura inicial 24
3.2. Nuestra figura con la construcción resultante 24
3.3. La primera parte del menú de creación de la macro 25
3.4. Segunda parte, los tres puntos ya están seleccionados 25
3.5. La tercera página, el círculo y su centro ya están seleccionados 26
3.6. La cuarta parte, el nombre y la descripción de la macro 26
3.7. El usuario selecciona los parámetros de entrada directamente en la figura . . 27
3.8. Una figura con tres puntos 27
3.9. La figura resultante con el círculo y su centro 27
3.10. Ejecutar una macro directamente desde el menú `Macro-constructions` . . . 28
3.11. La figura que obtendremos 31
3.12. Un polígono regular de 15 lados 49
3.13. Una curva fractal que simula la representación de un árbol 50
3.14. Ventana de diálogo para bloquear algunas herramientas de la interfaz 51
3.15. La ventana de diálogo para desbloquear la interfaz de una figura. 52

4.1. La ventana de diálogo de la sesión Dr. Geo 54
4.2. Una figura con un área de exportación definida 56
4.3. El área de exportación ha sido exportada a un documento PostScript; es visible con el programa GhostView 56

5.1. Trapecio Rectángulo 57
5.2. Teorema de Ptolomeo: cuadrilátero convexo 59
5.3. Teorema de Ptolomeo: cuadrilátero no convexo 59
5.4. Refutación de la conjetura 60
5.5. Construción de la raíz cuadrada de 2 60
5.6. Construcción de la raíz de 3 61
5.7. Espiral de Teodoro 62
5.8. La espiral de Baravelle obtenida al evaluar el código Scheme 63

5.9. Espiral simplificada 64
5.10. Cadena de Papus 65
5.11. Hexágono regular inscrito 66
5.12. Aproximación de π 66

6.1. Resultado de construir un polígono regular de 7 lados. 70
6.2. Ejemplo de párrafo de texto dentro de una figura 71

Índice alfabético

Ángulo
Geométrico, 16
Orientado, 16

Arco de círculo, 13
Longitud, 15
Ayuda
Preferencias predeterminadas
Navegador, 22

Círculo, 13
Perímetro, 15
Colocar
Texto, 71

Editar
Apariencia, 18
Estilo predeterminado (por default), 22
Preferencias predeterminadas
Nombres, 22
Preferencias predeterminadas (por default)
Deshacer/Hacer, 22
Preferencias predeterminadas(default), 21
Propiedades, 20
Punto, 20
Script, 20
Valor, 20
Enrejado
Activar, 21

Figura
Abrir, 54
Activar
Enrejado(rejilla), 21
Colocar
Texto, 71
Exportar
LaTeX, 54
Definir el área de exportación, 55
Fly Draw/WIMS, 55
PNG, 55
PostScript, 55
Guardar, 53
Imprimir, 70
Mover, 17
Figura Scheme de Dr. Geo
Creación de objetos, 39
Ángulo, 45
Arco, 42
Atributos(Apariencia) de los objetos, 46
Círculo, 41
Lugar Geométrico, 43
Número, 44
Polígono, 42
Punto, 39
Recta, 40
Segmento, 41
Semirrecta, 41
Transformaciones geométricas, 42
Vector, 43
Ejemplos, 36, 48
Cadena de Papus, 65
Espiral de Baravelle, 61
Espiral de Teodoro, 60
Fractal, 49
Polígono regular, 48
Introducción, 36
Nueva figura, 36
Sinónimos, 46
Figura Scheme de Dr. Geo
Evaluar, 36

Herramientas
Bloquear, 51
Desbloqueo, 52
Histórico
Árbol lógico, 17

Imprimir
Figura, 70

Línea, 13
Distancia, 15
Paralela, 14
Pendiente, 15
Perpendicular, 14
Locus, 14
Lugar Geométrico, 14

Lugar geométrico
- Script, 32

Macro, 17
- Abrir, 54
- Crear, 24
- Ejecutar, 26
  - Menu, 26
- Guardar, 53, 54
- Polígono, 69
- Script, 69

Número, 15
- Valor libre, 15

Objeto
- Borrar, 17
- Enmascarar, 18
- Mover, 17
- Ocultar, 18
- Renombrar, 18

Polígono, 14
- Macro, 69

Punto
- Coordenadas, 16
- Definido por sus coordenadas, 12
- Intersección, 12
- Libre, 12
- Medio, 12
- Renombrar, 18

Rayo, 13
Recta, 13
- Distancia, 15
- Ortogonal, 14
- Paralela, 14
- Pendiente, 15
- Perpendicular, 14

Rejilla
- Activar, 21

Rotación
- De 180 grados, 15

Script, 16
- `getAbscissa`, 33
- `getAngle`, 35
- `getCenter`, 34
- `getCoordinates`, 33
- `getLength`, 34, 35
- `getNormal`, 34
- `getNorm`, 34
- `getRadius`, 35
- `getSlope`, 34
- `getUnit`, 34
- `getValue`, 35
- `move`, 36
- `números aleatorios`, 30
- `setAbscissa`, 33
- `setCoordinates`, 34
- `setValue`, 35
- Con parámetros de entrada, 31
- Ejemplos, 29
  - Área de un trapecio, 57
  - Diferencia en la longitud de dos segmentos, 58
  - Perímetro, 58
  - Raíz cuadrada, 58
  - Tangente a una curva, 32
  - Teorema de Pitágoras, 58
  - Teorema de Ptolomeo, 58
- Intervalo, 31
- Introducción, 29

Segmento, 13
- Longitud, 15

Semirrecta, 13
Sesión
- Abrir, 54
- Guardar, 53

Texto
- colocar, 71

Transformación del plano
- Escala, 15
- Homotecia, 15
- Reflexión, 14
- Rotación
  - General, 15
- Simetría
  - Axial, 14
  - Central, 15
- Traslación, 15

Vector, 13
- Coordenadas, 16
- Longitud, 15
- Norma, 15

Apéndice A

Histórico.

Redacciónes originales de la documentación : Hilaire Fernandes (Inglés y Francés) y Andrea Centomo (Italiano).

Traducción al Español : Adrián Ulises Soto Bañuelos.

Apéndice B

GNU Free Documentation License

```
               GNU Free Documentation License
                 Version 1.2, November 2002


 Copyright (C) 2000,2001,2002  Free Software Foundation, Inc.
     59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other
functional and useful document "free" in the sense of freedom: to
assure everyone the effective freedom to copy and redistribute it,
with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way
to get credit for their work, while not being considered responsible
for modifications made by others.

This License is a kind of "copyleft", which means that derivative
works of the document must themselves be free in the same sense.  It
complements the GNU General Public License, which is a copyleft
license designed for free software.

We have designed this License in order to use it for manuals for free
software, because free software needs free documentation: a free
program should come with manuals providing the same freedoms that the
software does.  But this License is not limited to software manuals;
it can be used for any textual work, regardless of subject matter or
whether it is published as a printed book.  We recommend this License
principally for works whose purpose is instruction or reference.


1. APPLICABILITY AND DEFINITIONS
```

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple

HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on

the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications

adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,

you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number.

```
If the Document specifies that a particular numbered version of this
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation.  If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation.


ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

    Copyright (c)  YEAR  YOUR NAME.
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.2
    or any later version published by the Free Software Foundation;
    with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled "GNU
    Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the "with...Texts." line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.
```